MEMORY MANAGEMENT

M. M. ABORT TEST

1. KT11-D BASIC LOGIC TEST
2. KT11-D ACESS KEYS TEST
3. MTPI/MFPI W/MEMORY MANAGEMENT
4. KT11-D PROCESSOR STATES TEST
5. STACK LIMIT TEST STANDARD ON 11/45
   KJ11-4 OPTION ON 11/40
6. KT11-D EXERCISER
7. MEMORY MANAGEMENT ABORT TESTS

digital

# MAINDEC CHANGE NOTICE

143 MCN

CHANGE NO.
11-DBKTF-A - 1
Sheet 1 of 1

| AUTHOR | PROGRAM DATE | PRODUCT LINE | MAINDEC NUMBER |
|---|---|---|---|
| R. Whitton<br>DATE EXT.<br>12/6/72 4271 | SEPT. 72 | PDP-11<br>V20-06251 | 11-DBKTF-A |

PROGRAM NAME Memory Management Abort Test DEVICE KT11-D

| ITEM | |
|---|---|
| Ø. | Program released to library. |
| 1. | This patch should be used for systems with less than 28K of core. A location in the 24K block in used which is not necessary and will cause an error due to a non-existing memory location.<br>LOCATION NEW CONTENTS<br>4154 240<br>4156 240<br>4160 240 |

IDENTIFICATION

| | |
|---|---|
| PRODUCT CODE: | MAINDEC-11-D0KTA-A-D |
| PRODUCT NAME: | KT11-D BASIC LOGIC TEST |
| DATE CREATED: | SEPTEMBER 1, 1972 |
| MAINTAINER: | DIAGNOSTIC GROUP |
| AUTHOR: | ROBERT WHITTON |

COPYRIGHT © 1972
DIGITAL EQUIPMENT CORPORATION

1,0 ABSTRACT

THIS PROGRAM INCREMENTALLY TESTS THE BASIC LOGIC FUNCTIONS OF THE KT11-D MEMORY MANAGEMENT OPTION FOR THE PDP-11/40, THEY FULLY TEST RELOCATION, DIRECT AND INDIRECT ADDRESSING OF THE MEMORY MANAGEMENT REGISTERS, AND CORRECT OPERATION OF ALL THE BITS IN THE REGISTERS, THE VARIOUS ABORTS ARE TESTED, AS IS PROPER "LOCKING" AND "UNLOCKING" OF THE ERROR TRACKING LOGIC.

2,0 REQUIREMENTS

2,1 EQUIPMENT

PDP-11/40 WITH KT11-D OPTION

2,2 STORAGE

THE PROGRAM REQUIRES MEMORY LOCATIONS 0 TO 17474.

3,0 LOADING PROCEDURE

LOAD PROGRAM INTO MEMORY USING ABS LOADER.

4,0 STARTING PROCEDURE

LOAD ADDRESS 200.
SET DESIRED SWITCH REGISTER SETTINGS (ALL DOWN FOR WORST CASE).
PRESS START.
THE PROGRAM WILL DISPLAY THE NUMBER OF THE CURRENT SUBTEST IN THE DISPLAY REGISTER, AND WILL RING THE BELL ON COMPLETION OF A PASS.

5,0 OPERATING PROCEDURE

5,1 OPERATIONAL SWITCH SETTINGS

SW 15=1 OR UP -- HALT ON ERROR
SW 14=1 OR UP -- SCOPE LOOP
SW 13=1 OR UP -- INHIBIT PRINTOUT
SW 11=1 OR UP -- INHIBIT ITERATIONS
SW 10=1 OR UP-- HALT AT END OF CURRENT TEST
WITH NEXT TEST NUMBER IN DATA LIGHTS

5,2 SUBROUTINE ABSTRACTS

5,2,1 SCOPE

THIS SUBROUTINE CALL IS PLACED BETWEEN EACH SUBTEST, IT RECORDS THE STARTING ADDRESS OF EACH SUBTEST AS IT IS BEING ENTERED, IF A SCOPE LOOP IS REQUESTED, IT WILL JUMP TO THE START OF THE SUBTEST THAT THE SCOPE LOOP IS REQUESTED FOR, IF SCOPE LOOP IS NOT REQUESTED, THERE WILL BE 1024 ITERATIONS ON THAT SUBTEST BEFORE THE NEXT SUBTEST IS ENTERED, SWITCH 11 ON A 1 INHIBITS ITERATION OF SUBTESTS,

5,2,2 HLT

THIS EMT CALLS THE SUBROUTINE PRINT, WHICH PRINTS OUT THE LOCATION COUNTER AT THE TIME OF FAILURE AND THE CONTENTS OF THE PROCESSOR STATUS REGISTER, NOTE THAT THE LOCATION COUNTER WILL BE THE ADDRESS OF THE HLT PLUS TWO,

5,2,3 TRAPCATCHER

THIS IS A SERIES OF INSTRUCTIONS STARTING AT LOCATION 0 DESIGNED TO DETECT AND ISOLATE UNEXPECTED TRAPS AND INTERRUPTS TO THE TRAP AND INTERRUPT VECTOR AREA OF MEMORY,

EACH VECTOR ENTRANCE ADDRESS IS LOADED WITH THE ADDRESS OF THE NEXT LOCATION, THE NEXT LOCATION IS LOADED WITH A HALT (000000), THUS AN ILLEGAL TRAP OR INTERRUPT WILL CAUSE A HALT AT THE TRAP LOCATION PLUS TWO,

IF A HALT OCCURS IN THE TRAP OR INTERRUPT AREA EXAMINE REGISTER SIX, IT WILL CONTAIN THE CURRENT STACK ADDRESS, THE CONTENTS OF THE CURRENT STACK ADDRESS IS THE VALUE OF THE LOCATION COUNTER WHEN THE TRAP OR INTERRUPT OCCURRED,

5,2,4 EMTSRV (EMT DECODER)

THIS ROUTINE DECODES ALL EMT CALLS, INCLUDING PATCHES AND THE HLT CALL WHICH PASSES CONTROL TO THE PRINT ROUTINE.

5,2,5 CLRALL

THIS ROUTINE CLEARS ALL THE PAR/S AND PDR/S OF THE KT11-D, AS WELL AS SR0.

5,2,6 RWALL

THIS ROUTINE MAPS ALL PAGES TO BANK 0 BY CLEARING ALL THE PAR/S, ALL PAGES ARE MADE 4K READ-WRITE BY LOADING ALL THE PDR/S WITH THE VALUE 77406.

5,3 PROGRAM AND/OR OPERATOR ACTION

THE PROGRAM FIRST CHECKS THOSE PROPERTIES OF THE KT11-D WHICH CAN BE TESTED WITH MEMORY MANAGEMENT TURNED OFF. THEN, DESTINATION ONLY RELOCATION IS USED TO SHOW THAT BASIC RELOCATION IS WORKING CORRECTLY. FINALLY, FULL RELOCATION IS ENABLED AND MISCELLANEOUS ASPECTS OF THE KT11-D'S OPERATION ARE CHECKED.

6,0 ERRORS

6,1 ERROR PRINTOUT

PRINTOUTS ARE IN A STANDARD TWO-WORD FORMAT, THE FIRST WORD IS THE OCTAL VALUE OF THE PC+2 OF THE DETECTED ERROR, THE SECOND IS THE CONTENTS OF THE PROCESSOR STATUS REGISTER WHEN THE ERROR WAS DETECTED.

6,2 ERROR RECOVERY

IN GENERAL, TEST FAILURES WILL PRINTOUT AN ERROR MESSAGE AND CONTINUE. IF THE "HALT ON ERROR" SWITCH IS SET, HITTING CONTINUE WILL RECOVER. IF THE PROGRAM HANGS UP IN A LOOP, THE ERROR IS LIKELY TO BE A SIGNAL WHICH WAS NEVER RECEIVED. IF A HALT OCCURS IN THE TRAP AND VECTOR AREA THE PROGRAM MUST BE RESTARTED. IF THE PROGRAM HALTS IN THE MAIN FLOW, CONSULT THE LISTING IF NO MESSAGE IS TYPED OUT.

6,3 BRANCH SELF

A BRANCH TO SELF IS USED IN THE KT11-D DIAGNOSTICS TO INDICATED A FAILURE WHEN A HALT OR A HLT WORD TRAP CALL COULD LEAD TO PROBLEM.

7,0 RESTRICTIONS

PROGRAM MUST BE LOADED INTO LOWER 4K OF MEMORY.

8,0 MISCELLANEOUS

8,1 EXECUTION TIME

EACH PASS TAKES APPROXIMATELY 1 MINUTE WITH CORE MEMORY.

8,2 STACK POINTERS

THE KERNEL STACK POINTER IS USUALLY INITIALIZED TO 1000, HOWEVER, IN CERTAIN TESTS IT MAY BE INITIALIZED TO A LOWER ADDRESS (VIRTUAL) TO MAKE UP FOR RELOCATION OF THE BANK.

THE USER STACK POINTER IS INITIALIZED TO 400.

8.4 EXECUTION ORDER CHECKING

SINCE THE KT11-D MAY CAUSE AN INCORRECT FETCH IF IT IS NOT WORKING CORRECTLY, THE ORDER OF EXECUTION OF ALL SUBTESTS IS CHECKED. THE SCOPE ROUTINE, WHEN IT CHANGES FROM ONE SUBTEST TO THE NEXT, INCREMENTS A COUNTER CALLED TESTCT. AT THE START OF EACH SUBTEST, THIS COUNTER IS CHECKED FOR THE CORRECT VALUE FOR THAT SUBTEST. IF TESTS ARE NOT EXECUTED IN THE CORRECT ORDER, TESTCT WILL NOT CONTAIN THE EXPECTED VALUE, AND AN ERROR PRINTOUT WILL OCCUR.

9.0 PROGRAM DESCRIPTION

THE PROGRAM INITIALLY TESTS THOSE FEATURES OF THE KT11-D OPTION WHICH CAN BE TESTED WITHOUT TURNING ON MEMORY MANAGEMENT. IT THEN USES THE MAINTENANCE MODE (DESTINATION ONLY RELOCATION) TO TEST TURNING MEMORY MANAGEMENT ON AND OFF AND TO FULLY CHECK OUT RELOCATION. ONCE RELOCATION HAS BEEN FULLY TESTED, FULL PAGING IS USED TO TEST THE REMAINING OPERATIONS OF THE OPTION.

```
                        ;
                        ;BASIC LOGIC TEST OF THE KT11-D
                        ;COPYRIGHT 1972, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754

                        ;OPERATING INSTRUCTIONS
                        ;       1, LOAD TEST USING THE ABSOLUTE LOADER
                        ;       2, LOAD SA 200
                        ;       3, SET SR TO INITIAL SETTINGS
                        ;       4, PRESS START

                        ;SW15=1 CAUSES HALT ON ERROR
                        ;SW14=1 CAUSES SCOPE LOOPING
                        ;SW13=1 INHIBITS ERROR PRINTOUT
                        ;SW11=1 INHIBITS ITERATIONS
                        ;SW10=1 HALT AT END OF CURRENT TEST WITH TEST NUMBER IN DATA LIGHTS OF NEXT
                        ;               TEST, PRESS CONTINUE TO ADVANCE TO NEXT TEST, (WITH SW11=1)
                        ;DEFINITIONS
104400                          SCOPE=TRAP
000240                          NOP=240
000000                          R0=%0
000001                          R1=%1
000002                          R2=%2
000003                          R3=%3
000004                          R4=%4
000005                          R5=%5
000006                          R6=%6
000007                          R7=%7
000006                          SP=%6
000007                          PC=%7
177570                          SR=177570
177776                          PS=177776
177776                          STATUS=PS
104006                          HLT=104006
000001                          BIT0=1
000002                          BIT1=2
000004                          BIT2=4
000010                          BIT3=10
000020                          BIT4=20
000040                          BIT5=40
000100                          BIT6=100
000200                          BIT7=200
000400                          BIT8=400
001000                          BIT9=1000
002000                          BIT10=2000
004000                          BIT11=4000
010000                          BIT12=10000
020000                          BIT13=20000
040000                          BIT14=40000
100000                          BIT15=100000
```

```
                                ;LOAD TRAP CATCHER INTO 0 THRU 777
                                ;LOAD EACH VECTOR ADDRESS WITH THE ADDRESS OF THE NEXT
                                ;LOCATION, AND LOAD EACH LOCATION IMMEDIATELY FOLLOWING
                                ;A VECTOR ADDRESS WITH A HALT INSTRUCTION

                                ;LOAD VECTOR AREA
        000030                          .=30
000030  015770                          EMTSRV
000032  000340                          340
        000034                          .=34
000034  015200                          SCOPEC
000036  000000                          0

                                ;LOAD STARTING AREA
        000200                          .=200
000200  000167  000774                  JMP     START
        000210                          .=210
000210  000167  014666                  JMP     TESTX
                                ;LOAD DATA AREA
        000400                          .=400
000400  000000                  USTACK: 0
        001000                          .=.+376
001000  000000                  KSTACK: 0
001002  000000  000000  000000          .WORD 0,0,0,0
001010  000000
001012  123456                  K123:   123456
001014  134567                  K134:   134567
001016  177564                  TCSR:   177564
001020  177566                  TDBR:   177566
001022  000000                  TEMP:   0
                                ;
001024  177572                  SR0:    177572          ;KY11-D STATUS REGISTER ADDRESSES
001026  177573                  SR0H:   177573
001030  177574                  SR1:    177574
001032  177576                  SR2:    177576
```

```
                                        ;
001034                                  ADRTAB:
001034  177600                          UPDR0:  177600          ;USER PAGE DESCRIPTOR REGISTERS
001036  177602                          UPDR1:  177602
001040  177604                          UPDR2:  177604
001042  177606                          UPDR3:  177606
001044  177610                          UPDR4:  177610
001046  177612                          UPDR5:  177612
001050  177614                          UPDR6:  177614
001052  177616                          UPDR7:  177616
                                        ;
001054  177640                          UPAR0:  177640          ;USER PAGE ADDRESS REGISTERS
001056  177642                          UPAR1:  177642
001060  177644                          UPAR2:  177644
001062  177646                          UPAR3:  177646
001064  177650                          UPAR4:  177650
001066  177652                          UPAR5:  177652
001070  177654                          UPAR6:  177654
001072  177656                          UPAR7:  177656
                                        ;
001074  172300                          KPDR0:  172300          ;KERNEL PAGE DESCRIPTOR REGISTERS
001076  172302                          KPDR1:  172302
001100  172304                          KPDR2:  172304
001102  172306                          KPDR3:  172306
001104  172310                          KPDR4:  172310
001106  172312                          KPDR5:  172312
001110  172314                          KPDR6:  172314
001112  172316                          KPDR7:  172316
                                        ;
001114  172340                          KPAR0:  172340          ;KERNEL PAGE ADDRESS REGISTERS
001116  172342                          KPAR1:  172342
001120  172344                          KPAR2:  172344
001122  172346                          KPAR3:  172346
001124  172350                          KPAR4:  172350
001126  172352                          KPAR5:  172352
001130  172354                          KPAR6:  172354
001132  172356                          KPAR7:  172356
        001132                          ADREND= .=2
                                        ;
001134  177600                          PDRTAB: 177600          ;STARTING ADDRESSES OF PDR'S FOR EACH MODE
001136  172300                          PDREND: 172300
001140  177640                          PARTAB: 177640          ;STARTING ADDRESSES OF PAR'S FOR EACH MODE
001142  172340                                  172340

001144  001074                          STATAB: KPDR0           ;ADDRESS OF KERNEL TABLE OF PDR'S AND PAR'S
001146  000000                                  0
001150  001034                                  UPDR0           ;ADDRESS OF USER TABLE OF PDR'S AND PAR'S
001152  140000                          STAEND: 140000
```

```
001154  000000                    STAPNT: 0
001156  000000                    PASES:  0
001160  000000                    SAVEA:  0
001162  000000                    SAVEB:  0
001164  000250                    KTVEC:  250
001166  000252                    KTSTA:  252
001170  100361                    PDRM2:  100361
001172  000000                    FTITLE: 0
001174  000000                    TESTCT: 0
001176  000000                    BLOCKS: 0


                                  ;SET UP FOR START OF BASIC LOGIC TESTS
001200  005037  177776            START:  CLR     @#PS                    ;INITIALIZE STATUS
001204  012706  001000                    MOV     #KSTACK,SP              ;SETUP KERNEL STACK
001210  012767  002000  014074            MOV     #2000,ICOUNT            ;INITIALIZE ITERATION COUNT
001216  012767  001256  014072            MOV     #TEST1+2,RETURN         ;SETUP SCOPE AND ITERATION LOOP RETURN
001224  012767  000001  177742            MOV     #1,TESTCT               ;INITIALIZE TEST COUNT
001232  005767  177734                    TST     FTITLE                  ;DID TITLE PRINT
001236  001007                            BNE     TEST1+2                 ;YES, START TEST
001240  004767  014172                    JSR     PC,TYPE                 ;NO, PRINT TITLE
001244  014724                            MTIT
001246  005267  177720                    INC     FTITLE
001252  000401                            BR      .+4                     ;SKP SCOPE INSTRUCTION

                                  ;SR0 AND SR1 SHOULD BE INITIALIZED TO 0
001254  104400                  TEST1:  SCOPE
001256  012706  001000                    MOV     #KSTACK,SP              ;INITIALIZE KERNEL STACK POINTER
001262  004767  014552                    JSR     PC,ORDER                ;CHECK TEST SEQUENCE + INIT SR0
001266  000001                            1                               ;TEST NUMBER
001270  104006                            HLT                             ;TEST EXECUTED OUT OF SEQUENCE
001272  000005                            RESET                           ;ISSUE INIT
001274  005777  177524                    TST     @SR0                    ;CHECK SR0
001300  001401                            BEQ     .+4
001302  104006                            HLT                             ;SR0 WAS NOT INITIALIZED TO ZERO
001304  005777  177520                    TST     @SR1                    ;CHECK SR1
001310  001401                            BEQ     .+4
001312  104006                            HLT                             ;SR1 WAS NOT INITIALIZED TO ZERO
001314  012767  000010  013770            MOV     #10,ICOUNT              ;DROP ITERATION COUNT SINCE RESET IS USED

                                  ;CHECK READ/WRITE PROPERTIES OF ALL BITS IN SR0 EXCEPT 0 AND 8
                                  ;BY ROTATING A ONE THRU THE BIT POSITIONS BEING CHECKED
001322  104400                  TEST2:  SCOPE
001324  012706  001000                    MOV     #KSTACK,SP              ;INITIALIZE KERNEL STACK POINTER
001330  004767  014504                    JSR     PC,ORDER                ;CHECK TEST SEQUENCE + INIT SR0
001334  000002                            2                               ;TEST NUMBER
001336  104006                            HLT                             ;TEST EXECUTED OUT OF SEQUENCE
001340  005777  177460                    TST     @SR0                    ;CHECK SR0 INITIALLY
001344  001402                            BEQ     .+6
001346  104006                            HLT                             ;SR0 NOT ZERO AT START OF TEST
001350  000422                            BR      EXIT2
001352  012700  000001                    MOV     #1,R0                   ;R0 CONTAINS BIT INDICATING POSITION BEING TESTE
001356  010001                  LOOP2:  MOV     R0,R1
```

```
001360  010102                          MOV     R1,R2
001362  042701  000401                  BIC     #401,R1         ;DON'T SET THE BIT IN SR0 IF IT'S BIT 0 OR BIT 8
001366  042702  017777                  BIC     #17777,R2               ;CLEAR THE BIT IN R2 IF IT SHOULDN'T SET IN SR0
001372  010177  177426                  MOV     R1,@SR0
001376  020277  177422                  CMP     R2,@SR0                 ;CHECK SR0
001402  001421                          BEQ     .+4
001404  104006                          HLT                             ;SR0 INCORRECT WHEN VALUE IN R1
                                                                        ;WAS LOADED INTO IT
001406  006300                          ASL     R0
001410  103362                          BCC     LOOP2
001412  005077  177406                  CLR     @SR0
001416                          EXIT2:

                                ;BITS 0-11 OF ALL PAR'S SHOULD BE READ/WRITE
                                ;TEST BY ROTATING A BIT THRU EACH PAR
                                ;ALSO SHOWS THAT OUTPUT PATHS FROM PAR'S ARE OK
                                ;AND THAT EVERY PAR ADDRESS IS RESPONDED TO
001416  104400                  TEST3:  SCOPE
001420  012706  001000                  MOV     #KSTACK,SP              ;INITIALIZE KERNEL STACK POINTER
001424  004767  014410                  JSR     PC,ORDER                ;CHECK TEST SEQUENCE + INIT SR0
001430  000003                          3                               ;TEST NUMBER
001432  104006                          HLT                             ;TEST EXECUTED OUT OF SEQUENCE
001434  012767  002000  013650          MOV     #2000,ICOUNT            ;RESTORE ICOUNT
001442  004767  013346                  JSR     %7,CLRALL               ;INITIALIZE KT11-D REGISTERS
001446  012703  001140                  MOV     #PARTAB,R3              ;R3 POINTS TO TABLE OF PAR ADDRESSES
001452  012700  000002                  MOV     #2,R0                   ;R0 IS COUNTER OF STATES LEFT TO TEST
001456  012301                  LOOP3:  MOV     (R3)+,R1                ;PUT ADDRESS OF 1ST PAR IN SET IN R1
001460  012702  000010                  MOV     #10,R2                  ;R2 IS COUNTER OF PAR'S LEFT TO TEST IN SET
001464  012704  000001          LOOP3A: MOV     #1,R4                   ;R4 IS BIT OF PAR BEING TESTED
001470  010411                  LOOP3B: MOV     R4,@R1                  ;SET BIT IN PAR
001472  020411                          CMP     R4,@R1                  ;CHECK PAR
001474  001401                          BEQ     .+4                     ;BRANCH IF OK
001476  104006                          HLT                             ;PAR WHOSE ADDRESS IS IN R1
                                                                        ;FAILED WHEN THE VALUE IN R4
                                                                        ;WAS LOADED INTO IT
001500  006304                          ASL     R4
001502  020427  010000                  CMP     R4,#10000
001506  001370                          BNE     LOOP3B
001510  005011                          CLR     @R1
001512  005721                          TST     (R1)+                   ;MOVE POINTER
001514  077215                          SOB     R2,LOOP3A               ;TEST ALL PAR'S IN SET
001516  077021                          SOB     R0,LOOP3                ;TEST ALL 3 REGISTER SETS

                                ;BITS 1-3, 8-14 OF ALL PDR'S SHOULD BE READ/WRITE
                                ;BITS 0,4,5,7 AND 15 SHOULD ALWAYS BE ZERO
                                ;BIT 6  SHOULD BE ZERO IF PDR IS WRITTEN
                                ;ACTUAL CLEARING AND SETTING OF 6 TESTED LATER
                                ;ALSO SHOWS THAT OUTPUT PATHS FROM PDR'S ARE OK
                                ;AND THAT EVERY PDR ADDRESS IS RESPONDED TO
001520  104400                  TEST4:  SCOPE
001522  012706  001000                  MOV     #KSTACK,SP              ;INITIALIZE KERNEL STACK POINTER
001526  004767  014306                  JSR     PC,ORDER                ;CHECK TEST SEQUENCE + INIT SR0
001532  000004                          4                               ;TEST NUMBER
001534  104006                          HLT                             ;TEST EXECUTED OUT OF SEQUENCE
```

```
001536  004767  013252                  JSR     %7,CLRALL       ;INITIALIZE KT11-D REGISTERS
001542  012703  001134                  MOV     #PDRTAB,R3
001546  012301                  LOOP4:  MOV     (R3)+,R1
001550  012702  000010                  MOV     #10,R2          ;LOAD ADDRESS OF 1ST PDR IN STATE
                                                                ;USE R2 AS A COUNTER OF PDR'S
                                                                ;LEFT TO TEST
001554  012700  000001          LOOP4A: MOV     #1,R0           ;SETUP R0 TO ROTATE A BIT THRU
001560  010005                  LOOP4B: MOV     R0,R5
001562  046705  177402                  BIC     PDRM2,R5        ;R5 CONTAINS EXPECTED RESULTING CONTENTS OF PDR
001566  010011                          MOV     R0,@R1          ;LOAD PDR
001570  021105                          CMP     @R1,R5          ;CHECK RESULTING CONTENTS OF PDR
001572  001401                          BEQ     .+4
001574  104006                          HLT

                                                                ;PDR WHOSE ADDRESS IS IN R1
                                                                ;WAS INCORRECT AFTER VALUE IN R0
                                                                ;WAS LOADED INTO IT
001576  006300                          ASL     R0              ;ROTATE BIT
001600  103367                          BCC     LOOP4B          ;BRANCH IF NOT DONE WITH THIS PDR
001602  005011                          CLR     @R1             ;IF DONE WITH THIS PDR, CLEAR IT
001604  005721                          TST     (R1)+           ;MOVE POINTER TO ADDRESS NEXT PDR
001606  077216                          SOB     R2,LOOP4A       ;TEST ALL PDR'S IN THIS GROUP
001610  020327  001136                  CMP     R3,#PDREND      ;TEST ALL 2 GROUPS OF PDRS-USER, KERNEL
001614  003754                          BLE     LOOP4

                                ;NO DUAL ADDRESSING TEST FOR PAR'S AND PDR'S
001616  104400                  TEST5:  SCOPE
001620  012706  001000                  MOV     #KSTACK,SP      ;INITIALIZE KERNEL STACK POINTER
001624  004767  014210                  JSR     PC,ORDER        ;CHECK TEST SEQUENCE + INIT SR0
001630  000005                          5                       ;TEST NUMBER
001632  104006                          HLT                     ;TEST EXECUTED OUT OF SEQUENCE
001634  004767  013154                  JSR     %7,CLRALL       ;CLEAR ALL PAR'S AND PDR'S
001640  012701  001034                  MOV     #ADRTAB,R1      ;R1 POINTS TO ADDRESS OF LOCATION
                                                                ;LOADED WITH 1 BIT SET IN EACH 4 BITS
001644  012702  001034          LOP5AA: MOV     #ADRTAB,R2      ;R2 USED AS A POINTER TO CYCLE THRU
                                                                ;ALL OTHER ADDRESSES OF PAR/PDR PAIR'S TO
                                                                ;CHECK FOR DUAL ADDRESSING
001650  012703  000040                  MOV     #32.,R3         ;R3 USED AS A COUNTER
001654  012771  010421  000000          MOV     #10421,@(R1)    ;LOAD A PAR OR PDR - SET ONE BIT
                                                                ;IN EACH CHIP (4 BITS PER CHIP) IF R/W
001662  020201                  LOP5B:  CMP     R2,R1           ;SKIP CHECKING THIS ADDRESS TO SEE IF
001664  001406                          BEQ     CONT5           ;IT'S A DUAL, SINCE IT WAS THE ONE LOADED
001666  005772  000000                  TST     @(R2)           ;OTHERWISE, CHECK TO SEE IF THIS
                                                                ;REGISTER RESPONDED TO THE ADDRESS
001672  001403                          BEQ     CONT5           ;OF THE ONE LOADED AS A DUAL
001674  104006                          HLT                     ;BRANCH IF OK
                                                                ;DUAL ADDRESSING - ADDRESS POINTED
                                                                ;TO BY R2 RESPONDED TO THE ADDRESS
                                                                ;POINTED TO BY R1 IN AT LEAST ONE
001676  005072  000000                  CLR     @(R2)           ;4 BIT SECTION (1 CHIP)
001702  005722          CONT5:  TST     (R2)+           ;REINITIALIZE FAULTY LOCATION
001704  077312                  SOB     R3,LOP5B        ;MOVE POINTER R2
                                                                ;CHECK ALL PAR'S AND PDR'S
                                                                ;TO SEE IF THEY RESPONDED TO THE
00[redacted]  022701  001132                  CMP     #ADREND[redacted]  ;ADDRESS POINTED TO BY R1
                                                                ;HAVE ALL ADDRESSES BEEN CHECKED
                                                                ;FOR DUALS?
```

```
001712  001402                          BEQ     DONE5A                  ;YES - GO TO NEXT TEST
001714  005031                          CLR     @(R1)+                  ;NO - MOVE POINTER R1
001716  000752                          BR      LOP5AA                  ;CHECK TO SEE IF ANY OTHER ADDRESS
                                                                        ;ALSO RESPONDS TO THE ADDRESS POINTED
                                                                        ;TO BY R1
001720  012767  000100  013364  DONE5A: MOV     #100,;COUNT             ;DROP ITERATION COUNT

                                ;SHOW THAT BYTE ADDRESSING OF PAR'S WORKS FOR HIGH AND LOW BYTES
001726  104400                  TEST6:  SCOPE
001730  012706  001000                  MOV     #KSTACK,SP              ;;INITIALIZE KERNEL STACK POINTER
001734  004767  014100                  JSR     PC,ORDER                ;CHECK TEST SEQUENCE + INIT SR0
001740  000006                          6                               ;TEST NUMBER
001742  104006                          HLT                             ;TEST EXECUTED OUT OF SEQUENCE
001744  012767  002000  013340          MOV     #2000,;COUNT            ;RESTORE ITERATION COUNT
001752  004767  013036                  JSR     %7,CLRALL               ;INITIALIZE KT11-D REGISTERS
001756  012703  001140                  MOV     #PARTAB,R3              ;R3 POINTS TO TABLE OF PAR ADDRESSES
001762  012700  000002                  MOV     #2,R0                   ;R0 IS COUNTER OF STATES LEFT TO TEST
001766  012301                  LOOP6:  MOV     (R3)+,R1                ;PUT ADDRESS OF 1ST PAR IN SET IN R1
001770  012702  000010                  MOV     #10,R2                  ;R2 IS COUNTER OF PAR'S LEFT TO TEST IN SET
001774  012711  177777          LOOP6A: MOV     #-1,@R1                 ;SET UP PAR BEING TESTED
002000  105011                          CLRB    @R1                     ;CLEAR LOW BYTE OF PAR
002002  022711  007400                  CMP     #7400,@R1               ;CHECK PAR
002006  001401                          BEQ     .+4                     ;BRANCH IF OK
002010  104006                          HLT                             ;DATOB TO PAR WHOSE ADDRESS IS IN
                                                                        ;R1 FAILED
002012  012711  177777                  MOV     #-1,@R1                 ;SET UP PAR TO TEST HIGH BYTE
002016  105061  000001                  CLRB    1(R1)                   ;CLEAR HIGH BYTE
002022  022711  000377                  CMP     #377,@R1                ;CHECK PAR
002026  001401                          BEQ     .+4
002030  104006                          HLT                             ;DATOB TO HIGH BYTE OF PAR WHOSE
                                                                        ;ADDRESS IS IN R1 FAILED
002032  005721                          TST     (R1)+                   ;MOVE POINTER
002034  077221                          SOB     R2,LOOP6A               ;TEST ALL PAR'S IN SET
002036  077025                          SOB     R0,LOOP6                ;TEST ALL 2 REGISTER SETS

                                ;SHOW THAT BYTE ADDRESSING OF PDR'S WORKS FOR HIGH AND LOW BYTES
002040  104400                  TEST7:  SCOPE
002042  012706  001000                  MOV     #KSTACK,SP              ;;INITIALIZE KERNEL STACK POINTER
002046  004767  013766                  JSR     PC,ORDER                ;CHECK TEST SEQUENCE + INIT SR0
002052  000007                          7                               ;TEST NUMBER
002054  104006                          HLT                             ;TEST EXECUTED OUT OF SEQUENCE
002056  004767  012732                  JSR     %7,CLRALL               ;INITIALIZE KT11-D REGISTERS
002062  012703  001134                  MOV     #PDRTAB,R3              ;R3 POINTS TO TABLE OF PDR ADDRESSES
002066  012700  000002                  MOV     #2,R0                   ;R0 IS COUNTER OF STATES LEFT TO TEST
002072  012301                  LOOP7:  MOV     (R3)+,R1                ;PUT ADDRESS OF 1ST PDR IN SET INTO R1
002074  012702  000010                  MOV     #10,R2                  ;R2 IS COUNTER OF PDR'S LEFT TO TEST IN SET
002100  012711  177777          LOOP7A: MOV     #-1,@R1                 ;SET UP PDR BEING TESTED
002104  105011                          CLRB    @R1                     ;CLEAR LOW BYTE OF PDR
002106  022711  077400                  CMP     #77400,@R1              ;CHECK PDR
002112  001401                          BEQ     .+4                     ;BRANCH IF OK
002114  104006                          HLT                             ;DATOB TO PDR WHOSE ADDRESS IS
                                                                        ;IN R1 FAILED
002116  012711  177777                  MOV     #-1,@R1                 ;SET UP PDR TO TEST HIGH BYTE
002122  105061  000001                  CLRB    1(R1)                   ;CLEAR HIGH BYTE
```

```
002126  022711  000016                  CMP     #16,@R1                 ;CHECK PDR
002132  001401                          BEQ     .+4
002134  104006                          HLT                             ;DATOB TO HIGH BYTE OF PDR WHOSE
                                                                        ;ADDRESS IS IN R1 FAILED
002136  005721                          TST     (R1)+                   ;MOVE POINTER
002140  077221                          SOB     R2,LOOP7A               ;TEST ALL PDR'S IN SET
002142  077025                          SOB     R0,LOOP7                ;TEST ALL 2 REGISTER SETS

                                ;;INIT SHOULD HAVE NO EFFECT ON PAR'S
002144  104400                  TEST10: SCOPE
002146  012706  001000                  MOV     #KSTACK,SP              ;INITIALIZE KERNEL STACK POINTER
002152  004767  013662                  JSR     PC,ORDER                ;CHECK TEST SEQUENCE + INIT SR0
002156  000010                          10                              ;TEST NUMBER
002160  104006                          HLT                             ;TEST EXECUTED OUT OF SEQUENCE
002162  012767  000010  013122          MOV     #10,ICOUNT              ;DROP ITERATION COUNT
002170  005067  000104                  CLR     TST10F
002174  012704  005252                  MOV     #5252,R4
002200  012703  001140          TST10:  MOV     #PARTAB,R3
002204  012700  000002                  MOV     #2,R0
002210  012301                  LOOP10: MOV     (R3)+,R1
002212  012702  000010                  MOV     #10,R2                  ;COUNTER TO LOAD PAR'S
002216  010421                  LOP10A: MOV     R4,(R1)+                ;LOAD PAR WITH PATTERN
002220  077202                          SOB     R2,LOP10A               ;LOAD ALL 16 IN THIS SET
002222  077006                          SOB     R0,LOOP10               ;INITIALIZE ALL 2 SETS
002224  000005                          RESET                           ;ISSUE INIT
002226  012703  001140                  MOV     #PARTAB,R3
002232  012700  000002                  MOV     #2,R0
002236  012301                  LOP10B: MOV     (R3)+,R1
002240  012702  000010                  MOV     #10,R2                  ;COUNTER TO CHECK PAR'S
002244  020411                  LOP10C: CMP     R4,@R1                  ;CHECK DATA
002246  001401                          BEQ     .+4
002250  104006                          HLT                             ;PAR WHOSE ADDRESS IS IN R1
                                                                        ;WAS INCORRECT AFTER INIT
002252  005721                          TST     (R1)+                   ;MOVE POINTER
002254  077205                          SOB     R2,LOP10C               ;TEST ALL 8 PAR'S IN THIS SET
002256  077011                          SOB     R0,LOP10B               ;TEST ALL 2 REGISTER SETS
002260  005767  000014                  TST     TST10F                  ;CHECK FOR BOTH PATTERNS USED
002264  001006                          BNE     EXIT10                  ;IF DONE, GO TO NEXT TEST
002266  005267  000006                  INC     TST10F                  ;IF NOT, SET FLAG
002272  012704  002525                  MOV     #2525,R4                ;LOAD OTHER PATTERN
002276  000740                          BR      TST10                   ;REPEAT TEST WITH 2ND PATTERN
002300  000000                  TST10F: 0
002302                          EXIT10:

                                ;;INIT SHOULDN'T CLEAR OR SET ANY OF THE R/W BITS IN THE PDR'S
002302  104400                  TEST11: SCOPE
002304  012706  001000                  MOV     #KSTACK,SP              ;INITIALIZE KERNEL STACK POINTER
002310  004767  013524                  JSR     PC,ORDER                ;CHECK TEST SEQUENCE + INIT SR0
002314  000011                          11                              ;TEST NUMBER
002316  104006                          HLT                             ;TEST EXECUTED OUT OF SEQUENCE
002320  005067  000104                  CLR     TST11F
002324  012704  025012                  MOV     #25012,R4               ;LOAD PATTERN IN R4
00[redacted]  012703  001134          TST11:  MOV     #PDRTAB,R3
00[redacted]  012700  000002                  MOV     #2,R0
```

```
002340  012301                  LOOP11: MOV     (R3)+,R1
002342  012702  000010                  MOV     #10,R2
002346  010421                  LOP11A: MOV     R4,(R1)+        ;COUNTER TO LOAD PDR'S
002350  077202                          SOB     R2,LOP11A       ;LOAD PDR WITH PATTERN
002352  077006                          SOB     R0,LOOP11       ;LOAD ALL 8 IN THIS SET
002354  000005                          RESET                   ;INITIALIZE ALL 2 SETS
002356  012703  001134                  MOV     #PDRTAB,R3      ;ISSUE INIT
002362  012700  000002                  MOV     #2,R0
002366  012301                  LOP11B: MOV     (R3)+,R1
002370  012702  000010                  MOV     #10,R2          ;COUNTER TO CHECK PDR'S
002374  020411                  LOP11C: CMP     R4,@R1          ;CHECK DATA
002376  001401                          BEQ     .+4
002400  104006                          HLT                     ;PDR WHOSE ADDRESS IS IN R1
                                                                ;WAS INCORRECT AFTER INIT
002402  005721                          TST     (R1)+           ;MOVE POINTER
002404  077205                          SOB     R2,LOP11C       ;TEST ALL 8 PDR'S IN THIS SET
002406  077011                          SOB     R0,LOP11B       ;TEST ALL 2 REGISTER SETS
002410  005767  000014                  TST     TST11F          ;CHECK FOR BOTH PATTERNS USED
002414  001006                          BNE     EXIT11          ;IF DONE, GO TO NEXT TEST
002416  005267  000006                  INC     TST11F          ;IF NOT, SET FLAG
002422  012704  052404                  MOV     #52404,R4       ;LOAD 2ND PATTERN
002426  000740                          BR      TST11
002430  000000          TST11F: 0
002432  000240          EXIT11: NOP

                                ;SHOW THAT SR1 IS ONLY # 0 AND CANNOT BE LOADED
002434  104400          TEST12: SCOPE
002436  012706  001000                  MOV     #KSTACK,SP      ;INITIALIZE KERNEL STACK POINTER
002442  004767  013372                  JSR     PC,ORDER        ;CHECK TEST SEQUENCE + INIT SR0
002446  000012                          12                      ;TEST NUMBER
002450  104006                          HLT                     ;TEST EXECUTED OUT OF SEQUENCE
002452  012767  002000  012632          MOV     #2000,ICOUNT    ;RESTORE ITERATION COUNT
002460  012777  177777  176342          MOV     #-1,@SR1        ;TRY TO LOAD SR1
002466  005777  176336                  TST     @SR1
002472  001401                          BEQ     .+4
002474  104006                          HLT                     ;SR1 INCORRECT # SHOULD HAVE TRACKED
                                ;SR2 SHOULD CONTAIN ADDRESS OF LAST FETCH WITH KT11-D TURNED OFF
                                ;CHECK THAT ABORT FREEZES SR2
002476  104400          TEST13: SCOPE
002500  012706  001000                  MOV     #KSTACK,SP      ;INITIALIZE KERNEL STACK POINTER
002504  004767  013330                  JSR     PC,ORDER        ;CHECK TEST SEQUENCE + INIT SR0
002510  000013                          13                      ;TEST NUMBER
002512  104006                          HLT                     ;TEST EXECUTED OUT OF SEQUENCE
002514  017701  176312          AD13:   MOV     @SR2,R1         ;PICK UP SR2 # SHOULD CONTAIN ADDRESS
002520  022701  002514                  CMP     #AD13,R1        ;OF THIS INSTRUCTION
002524  001401                          BEQ     .+4
002526  104006                          HLT                     ;SR2 DID NOT CONTAIN FETCH ADDRESS
002530  052777  100000  176266  AD13A:  BIS     #BIT15,@SR#     ;SET NR ABORT
002536  000240                          NOP
002540  022777  002530  176264          CMP     #AD13A,@SR2     ;CHECK IF SR2 FROZE
002546  001401                          BEQ     .+4
002550  104006                          HLT                     ;SR2 NOT BEING DISABLED BY NR ABORT
002552  042777  100000  176244          BIC     #BIT15,@SR0     ;CLEAR NR ABORT
002560  052777  040000  176236  AD13B:  BIS     #BIT14,@SR0     ;SET PL ABORT
```

```
002566  000240                          NOP
002570  022777  002560  176234          CMP     #AD13B,@SR2             ;DID SR2 FREEZE
002576  001401                          BEQ     .+4
002600  104006                          HLT                             ;SR2 NOT BEING DISABLED BY PL ABORT
002602  042777  040000  176214          BIC     #BIT14,@SR0             ;CLEAR PL ABORT
002610  052777  020000  176206  AD13C:  BIS     #BIT13,@SR0             ;SET RO ABORT
002616  000240                          NOP
002620  022777  002610  176204          CMP     #AD13C,@SR2             ;DID SR2 FREEZE
002626  001401                          BEQ     .+4
002630  104006                          HLT                             ;SR2 NOT BEING DISABLED BY RO ABORT

                                ;SHOW THAT DESTINATION ONLY RELOCATION DOESN'T RELOCATE AN INSTRUCTION
                                ;FETCH (ONE CASE), AND THAT RESET CLEARS SR0<8>
                                ;AND TURNS OFF DESTINATION ONLY RELOCATION
                                ;IF THAT MUCH WORKS, YOU'LL GET THRU TO THE NEXT TEST
002632  104400                  TEST14: SCOPE
002634  012706  001000                  MOV     #KSTACK,SP              ;INITIALIZE KERNEL STACK POINTER
002640  004767  013174                  JSR     PC,ORDER                ;CHECK TEST SEQUENCE + INIT SR0
002644  000014                          14                              ;TEST NUMBER
002646  104006                          HLT                             ;TEST EXECUTED OUT OF SEQUENCE
002650  004767  012140                  JSR     %7,CLRALL               ;THIS TEST SHOULDN'T GO THRU ANY PAR/PDR PAIR'S
                                                                        ;SO MAKE THEM ALL GIVE NON-RESIDENT
                                                                        ;AND PAGE LENGTH ERRORS IF ACCESSED
002654  012777  001006  176212          MOV     #1006,@KPDR0            ;3 BLOCKS OF KERNEL PDR0 MUST BE MAPPED
                                                                        ;TO ALLOW TRAPS AND ABORTS
002662  012767  000010  012422          MOV     #10,ICOUNT              ;DROP THE ITERATION COUNT
002670  012777  000400  176126          MOV     #400,@SR0               ;TURN ON DESTINATION ONLY RELOCATION
002676  000005                          RESET                           ;SHOULD CLEAR DEST ONLY  BIT, AND A
                                                                        ;SOLID PLACE TO START
                                                                        ;IF THE FETCH IS RELOCATED
                                                                        ;THIS WILL GIVE A PL ABORT
002700  032777  000400  176116          BIT     #400,@SR0               ;IF KT11-D STILL ON, THIS SHOULD CAUSE
002706  001401                          BEQ     .+4                     ;PL AND NR ERRORS
002710  000000                          HALT                            ;IF KT11-D IS OFF, BIT 8 OF SR0 READS
                                                                        ;AS STILL SET OR ANOTHER BIT IS INCORRECT
                                                                        ;IF KT11-D IS ON, NO NR OR SL ABORT
                                                                        ;OCCURRED AND RESET FAILED TO TURN KT11-D OFF
002712  005077  176106                  CLR     @SR0

                                ;SHOW THAT DESTINATION ONLY RELOCATION DOESN'T RELOCATE THE SOURCE
                                ;ADDRESS AND DOES RELOCATE THE DESTINATION
002716  104400                  TEST15: SCOPE
002720  012706  001000                  MOV     #KSTACK,SP              ;INITIALIZE KERNEL STACK POINTER
002724  004767  013110                  JSR     PC,ORDER                ;CHECK TEST SEQUENCE + INIT SR0
002730  000015                          15                              ;TEST NUMBER
002732  104006                          HLT                             ;TEST EXECUTED OUT OF SEQUENCE
002734  012767  000010  012350          MOV     #10,ICOUNT              ;KEEP THE NUMBER OF LOOPS DOWN
002742  004767  012046                  JSR     %7,CLRALL
002746  012777  000001  176140          MOV     #1,@KPAR0               ;OFFSET KERNEL PAR/PDR PAIR 0 ONE BLOCK FROM BAN
002754  012777  077406  176112          MOV     #77406,@KPDR0
002762  012701  003034                  MOV     #DATA16,R1              ;LOAD A BANK 0 ADDRESS
002766  012777  000400  176030          MOV     #400,@SR0               ;TURN ON DESTINATION ONLY RELOCATION
003[illegible]  021111                  CMP     @R1,@R[illegible]       ;THIS TEST WILL FAIL IF BOTH ARE
003[illegible]  001001                  BNE     .+4                     ;RELOCATED OR BOTH ARE NOT RELOCATED
```

```
003000  000000                          HALT                    ;SOURCE AND DESTINATION BOTH ADDRESSED SAME LOCA
003002  000005                          RESET                   ;TURN OFF DESTINATION-ONLY RELOCATION
003004  012701  002734                  MOV     #DATA16-100,R1  ;LOAD DESTINATION ADDRESS MINUS RELOCATION FACTO
003010  012702  003034                  MOV     #DATA16,R2      ;LOAD SOURCE ADDRESS
003014  012777  000400  176002          MOV     #400,@SR0       ;TURN ON DESTINATION-ONLY RELOCATION
003022  021211                          CMP     @R2,@R1         ;USE SAME INSTRUCTION AND ADDRESS
003024  001401                          BEQ     .+4             ;MODES AS BEFORE
003026  000000                          HALT                    ;DESTINATION NOT RELOCATED OR INCORRECTLY
                                                                ;RELOCATED OR SOURCE RELOCATED
003030  000005                          RESET                   ;TURN OFF RELOCATION
003032  000401                          BR      .+4
003034  132465                  DATA16: 132465
                        ;SHOW THAT A DATO OF 0 TO BIT 8, SR0 THRU KERNEL PAGE 7 WILL
                        ;CLEAR THE DESTINATION ONLY RELOCATION BIT AND TURN OFF DESTINATION ONLY RELOCATION
003036  104400                  TEST16: SCOPE
003040  012706  001000                  MOV     #KSTACK,SP      ;INITIALIZE KERNEL STACK POINTER
003044  004767  012770                  JSR     PC,ORDER        ;CHECK TEST SEQUENCE & INIT SR0
003050  000016                          16                      ;TEST NUMBER
003052  104006                          HLT                     ;TEST EXECUTED OUT OF SEQUENCE
003054  004767  011734                  JSR     %7,CLRALL       ;INITIALIZE
003060  012777  000001  176026          MOV     #1,@KPAR0       ;MAP KERNEL PAR/PDR PAIR 0
                                                                ;TO BANK 0 OFFSET BY 1 PAGE
003066  012777  077406  176000          MOV     #77406,@KPDR0   ;USED TO PROVE KT11-D IS
                                                                ;TURNED OFF AFTER CLEARING BIT 8, SR0
003074  012701  003034                  MOV     #DATA16,R1      ;SETUP R1 TO REFERENCE KERNEL PAR/PDR PAIR 0
003100  004767  013020                  JSR     PC,KERN7        ;MAP KERNEL PAR/PDR 7 TO EXT BANK
003104  016702  175714                  MOV     SR0,R2          ;SETUP R2 TO ADDRESS SR0
003110  012777  000400  175706          MOV     #400,@SR0       ;TURN ON DESTINATION ONLY RELOCATION
003116  005012                          CLR     @R2             ;CLEAR SR0 THRU KERNEL PAR/PDR PAIR7
003120  021111                          CMP     @R1,@R1         ;SHOW THAT KT11-D IS OFF
003122  001401                          BEQ     .+4
003124  000000                          HALT                    ;KT11-D STILL ON
003126  032777  000400  175670          BIT     #400,@SR0       ;SHOW THAT BIT 8, SR0 IS NOW ZERO
003134  001402                          BEQ     .+6
003136  104006                          HLT                     ;DESTINATION ONLY RELOCATION BIT IS STILL ON
003140  000005                          RESET                   ;MAKE SURE THAT KT11-D IS OFF
                        ;SHOW THAT A DATO OF 0 TO BIT 8, SR0 THRU USER PAGE 7
                        ;WILL TURN OFF DESTINATION - ONLY PAGING
003142  004767  011646                  JSR     %7,CLRALL       ;INITIALLY CLEAR ALL PAR/PDR PAIRS
003146  012777  000001  175700          MOV     #1,@UPAR0       ;MAP USER 0 TO
003154  012777  077406  175652          MOV     #77406,@UPDR0   ;BANK 0 OFFSET BY 1 PAGE, RW
003162  012701  003034                  MOV     #DATA16,R1      ;SETUP R1 TO REFERENCE USER 0
003166  012777  007600  175676          MOV     #7600,@UPAR7    ;MAP USER 7 TO THE
003174  012777  077406  175650          MOV     #77406,@UPDR7   ;EXTERNAL BANK
003202  016702  175616                  MOV     SR0,R2          ;SETUP R2 TO ADDRESS SR0
003206  012737  140000  177776          MOV     #140000,@#PS    ;SET MODE TO USER
003214  012777  000400  175602          MOV     #400,@SR0       ;TURN ON DESTINATION - ONLY PAGING
003222  005012                          CLR     @R2             ;CLEAR SR0 THRU USER ASR7
003224  021111                          CMP     @R1,@R1         ;SHOW THAT KT11-D IS OFF
003226  001401                          BEQ     .+4
003230  000777                          BR      .               ;RELOCATION STILL ON

                        ;SHOW THAT ALL PAGE BOUNDARY REFERENCES REFERENCE THE CORRECT PAR
                        ;AND RELOCATE CORRECTLY
```

```
                                        ;USE DESTINATION = ONLY PAGING
                                        ;MAP ALL PAR/PDR PAIRS RESIDENT READ WRITE

                                        ;       R0      = POINTS TO THE ADDRESS OF THE CURRENT PAR IN THE ADDRESS TABLE
                                        ;       R1      = CONTAINS VIRTUAL ADDRESS BEING USED TO REFERENCE START OF PAGE
                                        ;       R2      = CONTAINS VIRTUAL ADDRESS BEING USED TO REFERENCE END OF PAGE
                                        ;       R3
                                        ;       R4
                                        ;       R5      = USED TO REFERENCE SR0 TO TURN OFF DESTINATION ONLY PAGING
003232  104400                  TEST17: SCOPE
003234  012706  001000                  MOV     #KSTACK,SP              ;INITIALIZE KERNEL STACK POINTER
003240  004767  012574                  JSR     PC,ORDER                ;CHECK TEST SEQUENCE & INIT SR0
003244  000017                          17                              ;TEST NUMBER
003246  104006                          HLT                             ;TEST EXECUTED OUT OF SEQUENCE
003250  004767  011540                  JSR     %7,CLRALL               ;INITIALIZE
003254  004767  011560                  JSR     %7,RWALL                ;MAKE ALL  PAR/PDR PAIRS RW, BANK 0,4K
003260  013767  017700  175672          MOV     @#17700,SAVEA           ;SAVE CONTENTS OF LOCATIONS TO BE USED
003266  013767  017776  175666          MOV     @#17776,SAVEB
003274  012737  123456  017700          MOV     #123456,@#17700         ;SET UP LOCATIONS TO BE REFERENCED
003302  012737  134567  017776          MOV     #134567,@#17776
003310  012703  001012                  MOV     #K123,R3
003314  012704  001014                  MOV     #K134,R4
003320  012767  000100  011764          MOV     #100,ICOUNT             ;CHANGE ITERATION COUNT
003326  012737  140000  177776          MOV     #140000,@#PS            ;CHANGE TO USER
003334  012706  000400                  MOV     #USTACK,SP              ;SET UP USER STACK POINTER
003340  005037  177776                  CLR     @#PS                    ;RETURN TO KERNEL
003344  012767  001144  175602          MOV     #STATAB,STAPNT          ;SET UP TO REFERENCE STATE TABLE
003352  017700  175576          STAT20: MOV     @STAPNT,R0              ;PICK UP ADDRESS OF START OF
003356  062700  000020                  ADD     #20,R0                  ;ADDRESS TABLE FOR NEW STATE
003362  062767  000002  175564          ADD     #2,STAPNT
003370  017737  175560  177776          MOV     @STAPNT,@#PS            ;SET UP NEW STATE
003376  062767  000002  175550          ADD     #2,STAPNT
003404  012767  000010  175544          MOV     #8.,PAGES               ;SET UP COUNTER OF ASR'S LEFT TO TEST
003412  012770  007600  000016          MOV     #7600,@16(R0)           ;SET UP SEGMENTED REFERENCE TO SR0,
003420  016705  175400                  MOV     SR0,R5                  ;USED TO TURN DESTINATION = ONLY PAGING OFF
003424  005001                          CLR     R1
003426  012702  000076                  MOV     #76,R2
003432  012767  000200  175536  PA620: MOV     #128.,BLOCKS            ;SET UP BLOCK COUNT
003440  012770  000177  000000          MOV     #177,@(R0)              ;SET UP PAR
003446  022767  000001  175502          CMP     #1,PAGES                ;IS THIS PAGE 7? (WAS USED
                                                                        ;FOR REFERENCE TO SR0)
003454  001005                          BNE     BLK20                   ;IF NOT, BRANCH
003456  012770  007600  177776          MOV     #7600,@-2(R0)           ;YES, SET UP PAGE 6 FOR REFERENCES TO SR0
003464  042705  020000                  BIC     #20000,R5               ;CHANGE R5 TO POINT TO SR0 THRU PAR/PDR PAIR6
003470  012777  000400  175326  BLK20: MOV     #400,@SR0               ;TURN ON DESTINATION ONLY PAGING
003476  021311                          CMP     @R3,@R1                 ;CK BOTTOM PAGE BOUNDARY
003500  001401                          BEQ     .+4
003502  000000                          HALT                            ;RELOCATION FAILED
003504  021412                          CMP     @R4,@R2                 ;CK UPPER PAGE BOUNDARY
003506  001401                          BEQ     .+4
003510  000000                          HALT                            ;RELOCATION FAILED
003512  005015                          CLR     @R5                     ;TURN OFF KT11-D
003514  005370  000000                  DEC     @(R0)                   ;MAP PAR 1 PAGE LOWER
003520  062701  000100                  ADD     #100,R1                 ;SET UP R1 AND R2 TO REFERENCE
```

```
003524  062702  000100                  ADD     #100,R2                 ;NEXT VIRTUAL PAGE
003530  005367  175442                  DEC     BLOCKS                  ;DECREMENT COUNT OF PAGES LEFT
003534  001355                          BNE     BLK20                   ;BRANCH IF NOT DONE WITH THIS PAR/PDR PAIR
003536  005070  000000                  CLR     @(R0)
003542  005367  175410                  DEC     PAGES                   ;DECREMENT COUNT OF PAR/PDR PAIR'S LEFT
003546  001402                          BEQ     END20                   ;BRANCH IF ALL PAR/PDR PAIR'S IN THIS STATE DONE
003550  005720                          TST     (R0)+                   ;MOVE ADDRESS TABLE POINTER
003552  000727                          BR      PAG20
003554  026727  175374  001152  END20:  CMP     STAPNT,#STAEND          ;CHECK FOR ALL STATES TESTED
003562  003673                          BLE     STAT20                  ;IF NOT, BRANCH
003564  005037  177776                  CLR     @#PS                    ;IF DONE, REINTIALIZE
003570  005077  175230                  CLR     @SR0
003574  016727  175360  017700          MOV     SAVEA,#17700
003602  016727  175354  017776          MOV     SAVEB,#17776
003610  016777  175352  175346          MOV     KTSTA,@KTVEC
003616  005077  175344                  CLR     @KTSTA

                                ;SHOW THAT THE INSTRUCTIONS USED IN THE NEXT TEST RELOCATE CORRECTLY IN
                                ;DESTINATION ONLY RELOCATION
003622  104400                  TEST20: SCOPE
003624  012706  001000                  MOV     #KSTACK,SP              ;INITIALIZE KERNEL STACK POINTER
003630  004767  012204                  JSR     PC,ORDER                ;CHECK TEST SEQUENCE & INIT SR0
003634  000020                          20                              ;TEST NUMBER
003636  104006                          HLT                             ;TEST EXECUTED OUT OF SEQUENCE
003640  012767  002000  011444          MOV     #2000,ICOUNT            ;RESTORE ITERATION COUNT
003646  004767  011142                  JSR     %7,CLRALL               ;CLEAR ALL KT11-D REGISTERS
003652  012777  000001  175234          MOV     #1,@KPAR0               ;OFFSET KERNEL I-SPACE PAGE 0
003660  012777  077406  175206          MOV     #77406,@KPDR0           ;BY 1 BLOCK FROM BANK 0
003666  004767  012232                  JSR     PC,KERN7                ;MAP KERNEL PAR/PDR 7 TO EXT BANK
003672  016701  175126                  MOV     SR0,R1                  ;SETUP R1 TO REFERENCE SR0
003676  016746  000054                  MOV     DST21A-100,-(SP)
003702  016746  000052                  MOV     DST21B-100,-(SP)
003706  016746  000050                  MOV     DST21C-100,-(SP)
003712  005067  000140                  CLR     DST21A                  ;INITIALIZE LOCATIONS TO BE
003716  012767  177777  000134          MOV     #-1,DST21B              ;WRITTEN INTO
003724  012767  177777  000130          MOV     #-1,DST21C
003732  012777  000400  175064          MOV     #400,@SR0               ;TURN ON DESTINATION & ONLY RELOCATION
003740  022737  176543  003642          CMP     #176543,@#AD21A-100     ;COMPARE THE CONTENTS OF AD21A
        003742                          AD21A=.-4                       ;WITH ITSELF, RELOCATED THRU KERNEL 0
003746  001401                          BEQ     .+4
003750  104006                          HLT                             ;DESTINATION & ONLY RELOCATION FAILED
                                                                        ;TO RELOCATE ONLY THE LAST CALCULATION
                                                                        ;OF THE CMP INSTRUCTION
003752  122737  165432  003654          CMPB    #165432,@#AD21B-100     ;COMPARE THE CONTENTS OF AD21B
        003754                          AD21B=.-4                       ;WITH ITSELF, RELOCATED THRU KERNEL 0
003760  001401                          BEQ     .+4
003762  104006                          HLT                             ;DESTINATION & ONLY RELOCATION
                                                                        ;FAILED TO RELOCATE ONLY THE FINAL
                                                                        ;CALCULATION OF THE CMPB INSTRUCTION
003764  012737  077711  003756          MOV     #77711,@#DST21A-100     ;EXECUTE REMAINING INSTRUCTIONS
003772  005077  000066                  CLR     @AD21C
003776  105037  003762                  CLRB    @#DST21C-100
004002  005011                          CLR     @R1                     ;TURN OFF KT11-D
004004  022767  077711  000044          CMP     #77711,DST21A           ;CHECK LOCATION ADDRESSED BY MOV
```

```
004012  001401                          BEQ     .+4
004014  104006                          HLT                             ;MOV INSTRUCTION FAILED TO RELOCATE
                                                                        ;ONLY THE FINAL ADDRESS CALCULATION
004016  005767  000036                  TST     DST21B                  ;CHECK LOCATION ADDRESSED BY CLR
004022  001401                          BEQ     .+4
004024  104006                          HLT                             ;CLR INSTRUCTION FAILED TO RELOCATE
                                                                        ;CORRECTLY IN DESTINATION - ONLY RELOCATION
004026  022767  177400  000026          CMP     #177400,DST21C          ;CHECK LOCATION ADDRESSED BY CLRB
004034  001401                          BEQ     .+4
004036  104006                          HLT                             ;CLRB INSTRUCTION FAILED TO RELOCATE
                                                                        ;CORRECTLY IN DESTINATION - ONLY RELOCATION
004040  012667  177716                  MOV     (SP)+,DST21C-100        ;RESTORE LOCATIONS IN CASE OF ERROR
004044  012667  177710                  MOV     (SP)+,DST21B-100
004050  012667  177702                  MOV     (SP)+,DST21A-100
004054  000404                          BR      EXIT21
004056  000000          DST21A: 0
004060  000000          DST21B: 0
004062  000000          DST21C: 0
004064  003760          AD21C:  DST21B-100
004066  000240          EXIT21: NOP

                                        ;TEST OF RELOCATION ADDERS - CHECK CORRECT PROPAGATION OF CARRY, AND CORRECT
                                        ;OUTPUT FOR EACH POSSIBLE COMBINATION FOR EACH BIT POSITION
                                        ;USE DESTINATION - ONLY RELOCATION, KERNEL
                                        ;TEST BY USING THE NECESSARY VALUE IN KERNEL PAR 1, WITH THE SECOND
                                        ;NECESPARY VALUE BEING THE VIRTUAL ADDRESS REFERENCE TO KERNEL PAR 1
                                        ;CHECK THE RESULTING PHYSICAL ADDRESS BY READING THE CONTENTS OF THE LOCATION,
                                        ;AND WRITING INTO THE LOCATION
                                        ;NOTE THAT THIS INCLUDES CHECKS OF ADDRESS WRAP AROUND
004070  104400          TEST21: SCOPE
004072  012706  001000                  MOV     #KSTACK,SP              ;INITIALIZE KERNEL STACK POINTER
004076  004767  011736                  JSR     PC,ORDER                ;CHECK TEST SEQUENCE + INIT SR0
004102  000021                          21                              ;TEST NUMBER
004104  104006                          HLT                             ;TEST EXECUTED OUT OF SEQUENCE
004106  004767  010702                  JSR     %7,CLRALL               ;CLEAR ALL KT11-D REGISTERS
004112  012777  077406  174754          MOV     #77406,@KPDR0           ;MAP KERNEL 0 TO BANK 0, 4K, RW
004120  012777  077406  174750          MOV     #77406,@KPDR1           ;MAKE KERNEL 1 4K, RW
004126  004767  011772                  JSR     PC,KERN7                ;MAP KERNEL PAR/PDR 7 TO EXT BANK

                                        ;CHECK VIRTUAL ADDRESS OF 0 ADDED TO PAR OF -1 (FOR BIT POSITIONS
                                        ;RELEVANT TO THE ADDERS ONLY)
004132  012777  007777  174756          MOV     #7777,@KPAR1            ;SET PAR TO -1
004140  012737  030000  177776          MOV     #30000,@#PS             ;SET UP LOCATION TO BE REFERENCED
004146  012777  000400  174650          MOV     #400,@SR0               ;TURN ON DESTINATION - ONLY PAGING
004154  122737  000060  020077          CMPB    #60,@#20077             ;CHECK HIGH BYTE OF RESULTING ADDRESS
                                                                        ;(PS) = REFERENCED THRU PAR/PDR PAIR 1
004162  001011                          BNE     ERR22A                  ;BRANCH ON FAILURE
004164  105037  020077                  CLRB    @#20077                 ;CLEAR PA 777777 THRU KERNEL 1
004170  005077  174630                  CLR     @SR0                    ;TURN OFF KT11-D
004174  105737  177777                  TSTB    @#PS+1                  ;CHECK TO SEE IF CORRECT LOCATION
004200  001401                          BEQ     .+4                     ;WAS REFERENCED
004202  104006                          HLT                             ;RELOCATION FAILED
004204  000405                          BR      CNT22B                  ;GO TO NEXT CHECK
004206  005077  174612          ERR22A: CLR     @SR0                    ;TURN OFF KT11-D
```

```
004212  104036                          HLT                             ;RELOCATION FAILED IN THE COMPARE
                                                                        ;AT LOCATION ADR22A
004214  005037  177776                  CLR     @#PS                    ;REINITIALIZE PROCESSOR STATUS

                                        ;CHECK VIRTUAL ADDRESS OF 1 ADDED TO PAR OF 0 (VALUES FOR BIT
                                        ;POSITIONS RELEVANT TO THE ADDERS ONLY), RESULT SHOULD BE PA 17712
004220  005077  174672          CNT22B: CLR     @KPAR1                  ;SET PAR TO 0
004224  012737  125252  017712            MOV     #125252,@#DESTAD        ;LOAD PHYSICAL LOCATION TO BE REFERENCED
                                                                        ;ADDRESS 17712
004232  012777  000400  174564            MOV     #400,@SR0               ;TURN ON DESTINATION = ONLY PAGING
004240  022737  125252  037712            CMP     #125252,@#37712         ;RELOCATE THRU KERNEL PAR/PDR PAIR1
004246  001011                          BNE     ERR22B                  ;BRANCH ON FAILURE
004250  005037  037712                  CLR     @#37712                 ;CLEAR THRU KERNEL PAR/PDR PAIR1
004254  005077  174544                  CLR     @SR0                    ;TURN OFF KT11-D
004260  005737  017712                  TST     @#17712                 ;CHECK TO SEE IF CORRECT LOCATION
004264  001401                          BEQ     .+4                     ;WAS CLEARED
004266  104006                          HLT                             ;RELOCATION FAILED
004270  000403                          BR      CNT22C                  ;GO TO NEXT CHECK
004272  005077  174526          ERR22B: CLR     @SR0                    ;TURN OFF KT11-D
004276  104006                          HLT                             ;RELOCATION FAILED IN THE COMPARE
                                                                        ;AT LOCATION ADR22B

                                        ;CHECK VIRTUAL ADDRESS OF 1 (BIT 6) ADDED TO PAR OF -1
                                        ;RESULTING PHYSICAL ADDRESS SHOULD BE ZERO
                                        ;NOTE THAT THIS IS A CHECK OF ADDRESS WRAP AROUND
004300  012777  007777  174610  CNT22C: MOV     #7777,@KPAR1            ;SET UP PAR TO -1
004306  012737  034343  000000            MOV     #34343,@#0              ;SET UP A VALUE IN LOCATION TO
                                                                        ;BE REFERENCED (0)
004314  012777  000400  174502            MOV     #400,@SR0               ;TURN ON DESTINATION-ONLY PAGING
004322  022737  034343  020100            CMP     #34343,@#20100          ;EFFECTIVELY ADDS 1 TO PAR ADDRESS
                                                                        ;TO GET PHYSICAL ADDRESS OF 0
004330  001013                          BNE     ERR22C                  ;BRANCH ON FAILURE
004332  012737  000002  020100            MOV     #2,@#20100              ;WRITE SAME LOCATION
004340  005077  174460                  CLR     @SR0                    ;TURN OFF KT11-D
004344  022737  000002  000000            CMP     #2,@#0                  ;CHECK LOCATION WHICH SHOULD HAVE
004352  001401                          BEQ     .+4                     ;BEEN REFERENCED
004354  104006                          HLT                             ;RELOCATION FAILED WHEN WRITING PA 0
004356  000406                          BR      CNT22D                  ;GO TO NEXT CHECK
004360  005077  174440          ERR22C: CLR     @SR0                    ;TURN OFF KT11-D
004364  104006                          HLT                             ;RELOCATION FAILED IN THE COMPARE
004366  012737  000002  000000            MOV     #2,@#0                  ;AT LOCATION ADR22C

                                        ;CHECK VIRTUAL ADDRESS OF -1 (BITS 6-12) ADDED TO PAR OF 1
                                        ;(PLUS HIGH BITS SET, BUT THEY DON'T ALTER CARRY CONDITION TESTED FOR)
                                        ;RESULTING PHYSICAL ADDRESS SHOULD BE ZERO
004374  012777  007601  174514  CNT22D: MOV     #7601,@KPAR1            ;SET UP PAR TO 1, WITH HIGH BITS SET
004402  012737  043434  000000            MOV     #43434,@#0              ;SET UP A VALUE IN LOCATION TO
                                                                        ;BE REFERENCED (0)
004410  012777  000400  174406            MOV     #400,@SR0               ;TURN ON DESTINATION-ONLY PAGING
004416  022737  043434  037700            CMP     #43434,@#37700          ;ALL HIGH BITS OF VA ARE 1, ADDED TO
                                                                        ;A ONE IN LOWEST BIT OF PAR TO PROPAGATE
                                                                        ;CARRY = RESULTING PHYSICAL ADDRESS 0
004424  001013                          BNE     ERR22D                  ;BRANCH ON FAILURE
004426  012737  000002  037700            MOV     #2,@#37700              ;WRITE SAME LOCATION
```

```
004434  005077  174364                  CLR     @SR0                    ;TURN OFF KT11-D
004440  022737  000002  000070          CMP     #2,@#0                  ;CHECK LOCATION WHICH SHOULD HAVE
                                                                        ;BEEN REFERENCED
004446  001401                          BEQ     .+4
004450  104006                          HLT                             ;RELOCATION FAILED WHEN WRITING PA 0
004452  000406                          BR      CNT22E                  ;GO TO NEXT CHECK
004454  005077  174344          ERR22D: CLR     @SR0                    ;TURN OFF KT11-D
004460  104006                          HLT                             ;RELOCATION FAILED IN THE COMPARE
                                                                        ;AT LOCATION ADR22D
004462  012737  000002  000000          MOV     #2,@#0                  ;RESTORE LOCATION REFERENCED

                                        ;CHECK VIRTUAL ADDRESS =1 ADDED TO PAR OF =1
                                        ;SHOULD GIVE RESULTING PA 17677
                                        ;NOTE THAT THIS IS A CASE OF ADDRESS WRAP AROUND
004470  012777  007777  174420  CNT22E: MOV     #7777,@KPAR1            ;SET UP PAR TO =1
004476  013746  017676                  MOV     @#17676,-(SP)           ;SAVE CONTENTS OF LOCATION TO BE
                                                                        ;REFERENCED
004502  012737  076767  017676          MOV     #76767,@#17676          ;LOAD LOCATION TO BE REFERENCED
004510  012777  000400  174306          MOV     #400,@SR0               ;TURN ON DESTINATION-ONLY PAGING
004516  122737  000175  037777          CMPB    #175,@#37777            ;READ LOCATION (VA=-1, PAR=-1)
                                                                        ;SHOULD GIVE PA 17677 (THRU KERNEL PAR1
004524  001012                          BNE     ERR22E                  ;BRANCH ON FAILURE
004526  105037  037777                  CLRB    @#37777                 ;WRITE SAME LOCATION
004532  005077  174266                  CLR     @SR0                    ;TURN OFF KT11-D
004536  022737  000367  017676          CMP     #367,@#17676            ;CHECK TO SEE IF CORRECT LOCATION
004544  001401                          BEQ     .+4                     ;WAS CLEARED (HIGH BYTE)
004546  104006                          HLT                             ;RELOCATION FAILED WHEN CLEARING
                                                                        ;PHYSICAL ADDRESS 17667 (THRU
004550  000403                          BR      END22E                  ;KERNEL PAR1)
004552  005077  174246          ERR22E: CLR     @SR0                    ;TURN OFF KT11-D
004556  104006                          HLT                             ;RELOCATION FAILED IN THE COMPARE AT
                                                                        ;LOCATION ADR22E
004560  012637  017676          END22E: MOV     (SP)+,@#17676           ;RESTORE LOCATION REFERENCED

                                        ;SHOW THAT SETTING SR0<0> TURNS ON FULL RELOCATION
                                        ;SHOW THAT ALL ADDRESS CALCULATIONS ARE RELOCATED
                                        ;SHOW THAT INIT CLEARS SR0<0> AND TURNS OFF RELOCATION
004564  104400                  TEST22: SCOPE
004566  012706  001000                  MOV     #KSTACK,SP              ;INITIALIZE KERNEL STACK POINTER
004572  004767  011242                  JSR     PC,ORDER                ;CHECK TEST SEQUENCE & INIT SR0
004576  000022                          22                              ;TEST NUMBER
004600  104006                          HLT                             ;TEST EXECUTED OUT OF SEQUENCE
004602  012767  000010  010502          MOV     #10,ICOUNT              ;DROP ITERATION COUNT
004610  004767  010200                  JSR     X7,CLRALL               ;INITIALLY CLEAR ALL KT11-D REGISTERS
004614  012777  000001  174272          MOV     #1,@KPAR0               ;MAP KERNEL PAGE 0 TO
004622  012777  077406  174244          MOV     #77406,@KPDR0           ;BANK 0 OFFSET BY 1 BLOCK
004630  004767  011270                  JSR     PC,KERN7                ;MAP KERNEL PAR/PDR 7 TO EXT BANK
004634  012767  052525  013050          MOV     #52525,DESTAD           ;INITIALIZE LOCATION  TO BE REFERENCED
004642  012777  000001  174154          MOV     #1,@SR0                 ;TURN ON RELOCATION
004650  000000                  ADD23:  HALT                            ;WITH RELOCATION ON, SHOULD FETCH
004652  000000                          HALT                            ;FROM 1 BLOCK ABOVE THIS
004654  000000                          HALT                            ;(ADD23A)
00[redacted]6  000000                  HALT
00[redacted]0  000000                  HALT
```

```
004662  000000                          HALT
004664  032777  000001  174132          BIT     #1,@SR0                 ;WHEN KT11-D IS TURNED OFF, NEXT
                                                                        ;FETCH SHOULD BE FROM HERE +
004672  001401                          BEQ     .+4                     ;CHECK BIT 0, SR0
004674  104006                          HLT                             ;KT11-D OFF BUT SR0<0> STILL SET
                                                                        ;AFTER AN INIT
004676  005077  174122                  CLR     @SR0
004702  000432                          BR      EXIT23
        004750                          .=ADD23+100
004750  022737  052525  017612          CMP     #52525,@#DESTAD-100     ;WHEN KT11-D IS TURNED ON, NEXT
                                                                        ;INSTRUCTION EXECUTED SHOULD
                                                                        ;BE HERE + CK RELOCATION OF SOURCE
004756  001401                          BEQ     .+4                     ;AND DESTINATION CALCULATIONS
004760  000000                          HALT                            ;RELOCATION FAILED IN A SOURCE OR
                                                                        ;DESTINATION ADDRESS CALCULATION
                                                                        ;WITH FULL RELOCATION ON (SR0<0>SET)
004762  000005                          RESET                           ;ISSUE INIT TO TURN OFF KT11-D
004764  000000                          HALT                            ;INIT DIDN'T TURN OFF KT11-D
004766  000777                          BR      .
004770  000240                  EXIT23: NOP

                                        ;SHOW THAT A DATO OF 0 TO SR0<0> WILL CLEAR SR0<0> AND
                                        ;TURN OFF RELOCATION
004772  104400                  TEST23: SCOPE
004774  012706  001000                  MOV     #KSTACK,SP              ;;INITIALIZE KERNEL STACK POINTER
005000  004767  011034                  JSR     PC,ORDER                ;CHECK TEST SEQUENCE + INIT SR0
005004  000023                          23                              ;;TEST NUMBER
005006  104006                          HLT                             ;;TEST EXECUTED OUT OF SEQUENCE
005010  012767  002000  010274          MOV     #2000,ICOUNT            ;RESTORE ITERATION COUNT
005016  004767  007772                  JSR     %7,CLRALL               ;;INITIALLY CLEAR ALL KT11-D REGISTERS
005022  012777  000001  174064          MOV     #1,@KPAR0               ;MAP KERNEL PAGE 0 TO
005030  012777  077406  174036          MOV     #77406,@KPDR0           ;BANK 0 OFFSET BY 1 BLOCK
005036  004767  011062                  JSR     PC,KERN7                ;MAP KERNEL PAR/PDR 7 TO EXT BANK
005042  012777  000001  173754          MOV     #1,@SR0                 ;TURN ON KT11-D
005050  000000                  ADD24:  HALT                            ;WHEN KT11-D IS TURNED ON, SHOULD
005052  000000                          HALT                            ;FETCH FROM THIS ADDRESS PLUS
005054  000000                          HALT                            ;ONE BLOCK (ADD24A)
005056  000240                          NOP
005060  000240                          NOP
005062  032777  000001  173734          BIT     #1,@SR0                 ;AFTER KT11-D IS TURNED OFF, CHECK
005070  001401                          BEQ     .+4                     ;SR0<0>
005072  104006                          HLT                             ;KT11-D OFF BUT SR0<0> STILL
                                                                        ;SET AFTER A BIC #1,@SR0
005074  000433                          BR      EXIT24

        005150                          .=ADD24+100
005150  042777  000001  173646          BIC     #1,@SR0                 ;WHEN KT11-D IS TURNED ON, SHOULD
                                                                        ;RELOCATE FETCH TO HERE + TURN
                                                                        ;OFF KT11-D VIA BIC OF SR0<0>
005156  000000                          HALT                            ;KT11-D STILL RELOCATING AFTER
005160  000005                          RESET                           ;BIC OF SR0<0>
005162  000777                          BR      .
005164  000240                  EXIT24: NOP
```

```
                                ;SHOW THAT A REFERENCE TO A NON-RESIDENT PAGE
                                ;WILL ABORT TO THE KT11-D ABORT VECTOR ADDRESS (250)
                                ;WITH BIT 15 OF SR0 SET,  SR0 AND SR2 ARE CHECKED FOR
                                ;THE CORRECT VALUES, AS ARE KPDR0 AND KPDR1
                                ;SHOW THAT BIT 15 OF SR0 CAN BE CLEARED
                                ;SHOW THAT SR2 IS READ ONLY

005166  104400                TEST24: SCOPE
005170  012706  001000                MOV     #KSTACK,SP      ;INITIALIZE KERNEL STACK POINTER
005174  004767  010640                JSR     PC,ORDER        ;CHECK TEST SEQUENCE + INIT SR0
005200  000024                        24                      ;TEST NUMBER
005202  104000                        HLT                     ;TEST EXECUTED OUT OF SEQUENCE
005204  004767  007604                JSR     %7,CLRALL       ;CLEAR ALL KT11-D REGISTERS
005210  012777  077406  173656        MOV     #77406,@KPDR0   ;MAP KERNEL 0 TO BANK 0, RW, 4K
005216  004767  010702                JSR     PC,KERN7        ;MAP KERNEL PAR/PDR 7 TO EXT BANK
005222  012777  005256  173734        MOV     #INT25,@KTVEC   ;SETUP RETURN VECTOR
005230  005077  173732                CLR     @KTSTA
005234  012704  020000                MOV     #20000,R4       ;USE R4 TO REFERENCE NR KERNEL 1
005240  005277  173560                INC     @SR0            ;TURN ON KT11-D
005244  005724                ADR25:  TST     (R4)+           ;REFERENCE NR KERNEL 1
005246  000000                ADR25A: HALT                    ;SHOULD HAVE ABORTED ALREADY
005250  005077  173550                CLR     @SR0            ;TURN OFF KT11-D
005254  000442                        BR      DON25
005256  017701  173542        INT25:  MOV     @SR0,R1         ;SAVE CONTENTS OF SR0
005262  005377  173536                DEC     @SR0            ;TURN OFF KT11-D
005266  022701  100003                CMP     #100003,R1      ;CHECK SAVED CONTENTS OF SR0
005272  001401                        BEQ     .+4
005274  104006                        HLT                     ;SR0 INCORRECT AFTER NR ABORT
                                                                ;(SEE SAVED CONTENTS IN R1)
005276  022777  005244  173526        CMP     #ADR25,@SR2     ;CK SR2
005304  001401                        BEQ     .+4
005306  104006                        HLT                     ;SR2 INCORRECT-SHOULD CONTAIN ADDRESS
                                                                ;OF LAST FETCH BEFORE THE ABORT
005310  005077  173516                CLR     @SR2            ;TRY TO WRITE INTO SR2
005314  022777  005244  173510        CMP     #ADR25,@SR2     ;SR2 SHOULD BE READ ONLY
005322  001401                        BEQ     .+4
005324  104006                        HLT
005326  022777  077506  173540        CMP     #77506,@KPDR0   ;SR2 NOT READ ONLY
005334  001401                        BEQ     .+4
005336  104006                        HLT

005340  005777  173532                TST     @KPDR1          ;KERNEL PDR 0 INCORRECT
005344  001401                        BEQ     .+4             ;W BIT SHOULD HAVE BEEN SET BY THE STACK WRITE
005346  104006                        HLT
005350  021627  005246                CMP     (R6),#ADR25A    ;KERNEL PDR 1 INCORRECT
005354  001401                        BEQ     .+4             ;CHECK VALUE PUSHED ON STACK
005356  104006                        HLT
005360  022626                        CMP     (R6)+,(R6)+     ;INCORRECT VALUE ON STACK
005362  005077  173600        DON25:  CLR     @KTSTA          ;RESTORE STACK
005366  016777  173574  173570        MOV     KTSTA,@KTVEC    ;CHANGE TRAP VECTOR TO CAUSE A
                                                                ;HALT ON A FALSE TRAP
                                ;SHOW THAT WRITING A PAGE WILL SET THE W BIT IN THE CORRESPONDING
                                ;PDR, AND THAT NO OTHER W BITS SET AT THE SAME TIME
                                ;SHOW THAT WRITING THE[illegible] (VIA A DATO) WILL CLEAR THE W BIT
                                ;SINCE THIS IS DONE FO[illegible]LL PDR'S, THIS IS ALSO
```

```
                                        ;A TEST OF INDIRECT ADDRESSING (VIA A VIRTUAL ADDRESS) OF THE PDR'S
005374  104400                  TEST25: SCOPE
005376  012706  001000                  MOV     #KSTACK,SP              ;INITIALIZE KERNEL STACK POINTER
005402  004767  010432                  JSR     PC,ORDER                ;CHECK TEST SEQUENCE & INIT SR0
005406  000025                          25                              ;TEST NUMBER
005410  104006                          HLT                             ;TEST EXECUTED OUT OF SEQUENCE
005412  012767  000400  007672          MOV     #400,ICOUNT             ;LOAD ITERATION COUNT
005420  004767  007414                  JSR     %7,RWALL                ;MAP ALL PAR/PDR PAIRS 4K, BANK 0, RW
005424  004767  010474                  JSR     PC,KERN7                ;MAP KERNEL PAR/PDR 7 TO EXT BANK
005430  012777  007600  173434          MOV     #7600,@UPAR7            ;MAP USER 7 TO EXTERNAL BANK
005436  012737  140000  177776          MOV     #140000,@#PS            ;SET MODE TO USER
005444  012706  000400                  MOV     #USTACK,R6              ;SET UP USER STACK
005450  005037  177776                  CLR     @#PS                    ;REINITIALIZE STATUS TO KERNEL MODE
005454  012704  001034                  MOV     #ADRTAB,R4
005460  012705  000010          LOP31A: MOV     #10,R5                  ;LOAD R4 WITH ADDRESS OF ADR TABLE
005464  022734  077406          LOP31B: CMP     #77406,@(R4)+           ;INIT COUNTER OF PDR'S LEFT TO CHECK
005470  001401                          BEQ     .+4                     ;CHECK ALL PDR W BITS BITS CLEAR
005472  104006                          HLT                             ;PDR INCORRECT - W BIT SET OR ANOTHER
                                                                        ;BIT INCORRECT IN PDR WHOSE ADDRESS
005474  077505                          SOB     R5,LOP31B               ;IS IN THE LOCATION POINTED TO BY R4
005476  062704  000020                  ADD     #20,R4                  ;MOVE POINTER TO FIRST ADR OF NEXT SET
005502  020427  001132                  CMP     R4,#ADREND
005506  003001                          BGT     CNT31A                  ;BRANCH IF DONE
005510  000763                          BR      LOP31A
005512  012700  001144          CNT31A: MOV     #STATAB,R0              ;SET UP START OF STATE TABLE
005516  012001          LOP31C: MOV     (R0)+,R1        ;R1 CONTAINS ADDRESS OF PDR OF ADDRESS
005520  012702  017776                  MOV     #17776,R2               ;SET UP VIRTUAL ADDRESS TO BE REFERENCED
005524  012037  177776                  MOV     (R0)+,@#PS              ;SET UP STATUS FOR CURRENT MODE
005530  005277  173270          LOP31D: INC     @$R0            ;TURN ON KT11-D
005534  011212                          MOV     (R2),(R2)               ;REFERENCE PAGE TO SET W BIT
005536  005077  173262                  CLR     @$R0                    ;TURN OFF KT11-D
005542  032771  000100  000000          BIT     #100,@(R1)              ;CHECK W BIT
005550  001001                          BNE     .+4
005552  104006                          HLT
                                                                        ;W BIT NOT SET IN PDR AFTER PAGE WRITTEN
005554  012703  001034                  MOV     #ADRTAB,R3              ;SET UP ADDRESS OF ADDRESS TABLE
005560  012704  000010                  MOV     #10,R4                  ;NOW CHECK ALL PDR TO SHOW NO OTHER
005564  020103          LOP31F: CMP     R1,R3                   ;W BITS WERE SET
005566  001405                          BEQ     CNT31B
005570  032773  000100  000000          BIT     #100,@(R3)
005576  001401                          BEQ     .+4
005600  104006                          HLT
                                                                        ;W BIT SET IN THE PDR WHOSE ADDRESS IS POINTED T
                                                                        ;AS WELL AS THE W BIT IN THE PDR
                                                                        ;FOR THE PAGE THAT WAS WRITTEN
005602  005723          CNT31B: TST     (R3)+                   ;UPDATE ADDRESS POINTER
005604  077411                          SOB     R4,LOP31F               ;TEST NEW PDW
005606  062703  000020                  ADD     #20,R3                  ;UPDATE POINTER TO NEXT SET
005612  020327  001132                  CMP     R3,#ADREND
005616  002760                          BLT     LOP31E
005620  012771  077406  000000          MOV     #77406,@(R1)            ;CLEAR W BIT VIA DATO TO PDR
005626  032771  000100  000000          BIT     #100,@(R1)              ;CHECK W BIT
005634  001401                          BEQ     .+4
005636  104006                          HLT
                                                                        ;W BIT DIDN'T CLEAR WHEN PDR
                                                                        ;WAS WRITTEN (ADDRESS OF ADDRESS
                                                                        ;OF PDR IS IN R1)
```

```
005640  005721                          TST     (R1)+                   ;UPDATE POINTER
005642  062702  020000                  ADD     #20000,R2               ;CHANGE VA TO REFERENCE NEXT PAGE
005646  103330                          BCC     LOP31D                  ;BRANCH TO TEST NEXT PAGE IN THIS MODE
005650  020027  001152                  CMP     R0,#STAEND              ;IF DONE THIS MODE, CHECK NEXT MODE
005654  002720                          BLT     LOP31C                  ;LOOP UNTIL ALL STATES HAVE BEEN TESTED
005656  005077  173142                  CLR     @SR0                    ;REINITIALIZE SR0

                                ;SHOW THAT A REFERENCE TO A NR PAGE WILL SET BOTH THE NR AND PL
                                ;ERROR BITS IF IT IS OUTSIDE THE MAPPED PAGE LENGTH
005662  104400                  TEST26: SCOPE
005664  012706  001000                  MOV     #KSTACK,SP              ;INITIALIZE KERNEL STACK POINTER
005670  004767  010144                  JSR     PC,ORDER                ;CHECK TEST SEQUENCE + INIT SR0
005674  000026                          26                              ;TEST NUMBER
005676  104006                          HLT                             ;TEST EXECUTED OUT OF SEQUENCE
005700  004767  007134                  JSR     %7,RWALL                ;MAP ALL PAGES RW,4K,BANK 0
005704  012777  000004  173164          MOV     #4,@KPDR1               ;MAP KERNEL 1 NR, 5 PAGE
005712  004767  010206                  JSR     PC,KERN7                ;MAP KERNEL PAR/PDR 7 TO EXT BANK
005716  012777  005742  173240          MOV     #RET33,@KTVEC           ;SETUP ABORT RETURN
005724  005077  173236                  CLR     @KTSTA
005730  005277  173070                  INC     @SR0                    ;TURN ON KT11-D
005734  005737  030000                  TST     @#30000                 ;REFERENCE NR KERNEL 1 - SHOULD ABORT
005740  000000                          HALT                            ;NO NR ABORT
005742  022777  140003  173054  RET33: CMP     #140003,@SR0            ;CHECK SR0
005750  001401                          BEQ     .+4
005752  104006                          HLT                             ;SR0 INCORRECT - SHOULD SHOW KERNEL
                                                                        ;PAGE 1, AND BOTH NR + PL ERRORS SET
005754  005077  173044                  CLR     @SR0
005760  016777  173202  173176          MOV     KTSTA,@KTVEC            ;RESTORE TRAP CATCHER

                                ;SHOW THAT KERNEL AND USER STACKS ARE ACCESSED CORRECTLY, AN IOT IS DONE TO
                                ;EACH MODE, THE LOCATION WRITTEN INTO WHEN THE STACK IS PUSHED
                                ;SHOWS WHICH STACK WAS USED.
005766  104400                  TEST27: SCOPE
005770  012706  001000                  MOV     #KSTACK,SP              ;INITIALIZE KERNEL STACK POINTER
005774  004767  010040                  JSR     PC,ORDER                ;CHECK TEST SEQUENCE + INIT SR0
006000  000027                          27                              ;TEST NUMBER
006002  104006                          HLT                             ;TEST EXECUTED OUT OF SEQUENCE
006004  004767  007004                  JSR     %7,CLRALL               ;INITIALIZE ALL KT11-D REGISTERS
006010  012706  000500                  MOV     #500,SP                 ;SET THE KERNEL STACK TO VIRTUAL ADDRESS 500
006014  012737  140000  177776          MOV     #140000,@#PS
006022  012706  000100                  MOV     #100,SP                 ;SET THE USER STACK TO VA 100
006026  005037  177776                  CLR     @#PS
006032  012777  077406  173034          MOV     #77406,@KPDR0           ;MAP KERNEL, AND USER TO BANK 0,4K,RW
006040  012777  077406  172766          MOV     #77406,@UPDR0
006046  012737  006114  000020          MOV     #KRET34,@#20            ;TEST USING IOT TRAP (THRU KERNEL SPACE)
006054  005037  000022                  CLR     @#22                    ;RETURN FROM TRAP IN KERNEL MODE
006060  016701  172740                  MOV     SR0,R1                  ;REFERENCE SR0 THRU R1
006064  004767  010034                  JSR     PC,KERN7                ;MAP KERNEL PAR/PDR 7 TO EXT BANK
006070  012777  077406  172754          MOV     #77406,@UPDR7           ;MAP USER PAGE 7
006076  012777  007600  172766          MOV     #7600,@UPAR7            ;TO THE EXTERNAL BANK
006104  005277  172714                  INC     @SR0                    ;TURN ON KT11-D
006110  000004                          IOT                             ;SHOULD USE STACK IN KERNEL ADDRESS SPACE
00[redacted]  000240                    NOP
00[redacted]  005011          KRET34: CLR     @R1                     ;TURN OFF KT11-D
```

```
006116  012737  006150  000020          MOV     #URET34,@#20            ;SETUP FOR IOT TO USER
006124  012737  140000  000022          MOV     #140000,@#22
006132  012737  140000  177776          MOV     #140000,@#PS
006140  005277  172660                  INC     @SR0                    ;TURN ON KT11-D
006144  000004                          IOT                             ;SHOULD USE STACK IN USER SPACE
006146  000240                          NOP
006150  005011                  URET34: CLR     @R1                     ;TURN OFF KT11-D
006152  022737  006112  000474          CMP     #KRET34-2,@#474
006160  001401                          BEQ     .+4
006162  104006                          HLT                             ;KERNEL STACK CONTENTS WRONG, PC NOT WHERE IT
006164  022737  000000  000476          CMP     #0,@#476                ;SHOULD HAVE BEEN PUSHED OR
006172  001401                          BEQ     .+4                     ;VALUE WRONG
006174  104006                          HLT                             ;KERNEL STACK WRONG-TRAP STATUS NOT
006176  022737  006146  000074          CMP     #URET34-2,@#74          ;NOT WHERE IT SHOULD HAVE BEEN PUSHED
006204  001401                          BEQ     .+4                     ;OR VALUE WRONG
006206  104006                          HLT                             ;USER STACK WRONG-PC NOT WHERE
006210  022737  140000  000076          CMP     #140000,@#76            ;IT SHOULD HAVE BEEN PUSHED
006216  001401                          BEQ     .+4                     ;OR VALUE WRONG
006220  104006                          HLT                             ;USER STACK WRONG-TRAP STATUS
                                                                        ;NOT WHERE IT SHOULD HAVE BEEN
                                                                        ;PUSHED OR VALUE WRONG
006222  012737  000076  000074          MOV     #76,@#74                ;REINITIALIZE LOCATIONS CHECKED
006230  005037  000076                  CLR     @#76
006234  012737  000476  000474          MOV     #476,@#474
006242  005037  000476                  CLR     @#476
006246  012706  001000                  MOV     #KSTACK,SP

                                ;SHOW THAT TRAP,EMT, AND INTERRUPTS TAKE VECTORS FROM KERNEL
                                ;IRREGARDLESS OF THE MODE AT THE TIME OF THE TRAP SEQUENCE
                                ;ALSO SHOW THAT ODD-ADDRESS TRAP (AN "INTERNAL"
                                ;TRAP) TAKES ITS VECTOR FROM KERNEL
                                ;NOTE THAT IF DUAL ADDRESSING OCCURS, THE ERROR
                                ;ADDRESS WILL BE USED (THE @ OVERRIDES THE 1)
006252  104400                  TEST30: SCOPE
006254  012706  001000                  MOV     #KSTACK,SP              ;INITIALIZE KERNEL STACK POINTER
006260  004767  007554                  JSR     PC,ORDER                ;CHECK TEST SEQUENCE & INIT SR0
006264  000030                          30                              ;TEST NUMBER
006266  104006                          HLT                             ;TEST EXECUTED OUT OF SEQUENCE
006270  005077  172530                  CLR     @SR0
006274  004767  006540                  JSR     %7,RWALL                ;MAP ALL PAR/PDR PAIRS RW, 4K, BANK 0
006300  012777  000001  172546          MOV     #1,@UPAR0               ;OFFSET USER 0 1 PAGE
006306  004767  007612                  JSR     PC,KERN7                ;MAP KERNEL PAR/PDR 7 TO EXT BANK
006312  012777  007600  172552          MOV     #7600,@UPAR7            ;MAP USER 7 TO THE EXTERNAL BANK
006320  016701  172500                  MOV     SR0,R1                  ;SETUP R1 TO REFERENCE SR0
006324  012737  140000  177776          MOV     #140000,@#PS            ;SETUP USER STACK
006332  012706  000400                  MOV     #USTACK,SP
006336  005037  177776                  CLR     @#PS
006342  012706  001000                  MOV     #KSTACK,SP              ;SETUP THE KERNEL STACK POINTER
006346  012737  006506  000130          MOV     #NG30B,@#130            ;SETUP SUCCESS RETURN
006354  012737  006524  000030          MOV     #OK30B,@#30             ;SETUP FAILURE RETURN
006362  005037  000132                  CLR     @#132
006366  005037  000032                  CLR     @#32
006372  012737  140000  177776          MOV     #140000,@#PS            ;SET MODE TO USER
006400  005277  172420                  INC     @SR0                    ;TURN ON KT11-D
```

```
006404  000000                          0
006406  000000                          0
006410  000000                          0
006412  000000                          0
006414  000000                          0
006416  000000                          0
006420  000000                          0
006422  000000                          0
006424  000000                          0
006426  000000                          0
006430  000000                          0
006432  000000                          0
006434  000000                          0
006436  000000                          0
006440  000000                          0
006442  000000                          0
006444  000000                          0
006446  000000                          0
006450  000000                          0
006452  000000                          0
006454  000000                          0
006456  000000                          0
006460  000000                          0
006462  000000                          0
006464  000000                          0
006466  000000                          0
006470  000000                          0
006472  000000                          0
006474  000000                          0
006476  000000                          0
006500  000000                          0
006502  000000                          0
006504  104000                          EMT
006506  022626                  NG35B:  CMP     (SP)+,(SP)+     ;SHOULD PICK UP RETURN ADDRESS FROM KERNEL
006510  005011                          CLR     @R1             ;RESTORE STACK POINTER
006512  012737  015770  000030          MOV     #EMTSRV,@#30    ;TURN OFF KT11-D
006520  104006                          HLT                     ;EMT VECTOR DIDN'T GO THRU KERNEL
006522  000402                          BR      IOT35
006524  022626                  OK35B:  CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
006526  005011                          CLR     @R1             ;TURN OFF KT11-D
006530  012737  015770  000030  IOT35:  MOV     #EMTSRV,@#30    ;RESTORE EMT SERVICE POINTER
006536  012737  000340  000032          MOV     #340,@#32
006544  012737  000132  000130          MOV     #132,@#130
006552  005037  000132                  CLR     @#132
006556  012737  006716  000120          MOV     #NG35C,@#120    ;SETUP IOT SUCCESS RETURN
006564  012737  006726  000020          MOV     #OK35C,@#20     ;SETUP IOT FAILURE RETURN
006572  005037  000122                  CLR     @#122
006576  005037  000022                  CLR     @#22
006602  012737  140000  177776          MOV     #140000,@#PS    ;SET MODE TO USER
006610  005277  172210                  INC     @SR0            ;TURN ON KT11-D
006614  000000                          0
006616  000000                          0
006620  000000                          0
006622  000000                          0
```

```
006624  000000                          0
006626  000000                          0
006630  000000                          0
006632  000000                          0
006634  000000                          0
006636  000000                          0
006640  000000                          0
006642  000000                          0
006644  000000                          0
006646  000000                          0
006650  000000                          0
006652  000000                          0
006654  000000                          0
006656  000000                          0
006660  000000                          0
006662  000000                          0
006664  000000                          0
006666  000000                          0
006670  000000                          0
006672  000000                          0
006674  000000                          0
006676  000000                          0
006700  000000                          0
006702  000000                          0
006704  000000                          0
006706  000000                          0
006710  000000                          0
006712  000000                          0
006714  000004                          IOT                             ;SHOULD PICK UP RETURN ADDRESS FROM KERNEL
006716  022626                  NG35C:  CMP     (SP)+,(SP)+             ;RESTORE STACK POINTER
006720  005011                          CLR     @R1                     ;TURN OFF KT11-D
006722  104006                          HLT                             ;TRAP VECTOR DIDN'T GO THRU KERNEL
006724  000402                          BR      INT35
006726  022626                  OK35C:  CMP     (SP)+,(SP)+             ;RESTORE STACK POINTER
006730  005011                          CLR     @R1                     ;TURN OFF KT11-D
006732  012737  000022  000020  INT35:  MOV     #22,@#20
006740  005037  000022                  CLR     @#22
006744  012737  000122  000120          MOV     #122,@#120
006752  005037  000122                  CLR     @#122
006756  012737  007140  000164          MOV     #NG35D,@#164            ;SETUP TTY SUCCESS RETURN
006764  012737  007154  000064          MOV     #OK35D,@#64             ;SETUP TTY FAILURE RETURN
006772  005037  000166                  CLR     @#166
006776  005037  000066                  CLR     @#66
007002  012737  140000  177776          MOV     #140000,@#PS            ;SET MODE TO USER
007010  005277  172010                  INC     @SR0                    ;TURN ON KT11-D
007014  000000                          0
007016  000000                          0
007020  000000                          0
007022  000000                          0
007024  000000                          0
007026  000000                          0
007030  000000                          0
007032  000000                          0
007034  000000                          0
```

```
007036  000000                          .
007040  000000                          .
007042  000000                          .
007044  000000                          .
007046  000000                          .
007050  000000                          .
007052  000000                          .
007054  000000                          .
007056  000000                          .
007060  000000                          .
007062  000000                          .
007064  000000                          .
007066  000000                          .
007070  000000                          .
007072  000000                          .
007074  000000                          .
007076  000000                          .
007100  000000                          .
007102  000000                          .
007104  000000                          .
007106  000000                          .
007110  000000                          .
007112  000000                          .
007114  012737  000100  177564          MOV     #100,@#177564           ;SET TTY INTERRUPT ENABLE-SHOULD
007122  000240                          NOP                             ;INTERRUPT IMMEDIATELY
007124  000240                          NOP
007126  005011                          CLR     @R1                     ;TURN OFF KT11-D
007130  005077  171662                  CLR     @TCSR                   ;CLEAR TTY IE
007134  104006                          HLT                             ;TTY FAILED TO INTERRUPT
007136  000412                          BR      ODDAD
007140  022626                  NG35D:  CMP     (SP)+,(SP)+             ;RESTORE STACK POINTER
007142  005011                          CLR     @R1                     ;TURN OFF KT11-D
007144  005077  171646                  CLR     @TCSR                   ;CLEAR TTY IE
007150  104006                          HLT                             ;TTY INTERRUPT DIDN'T GO THRU KERNEL
007152  000404                          BR      ODDAD
007154  022626                  OK35D:  CMP     (SP)+,(SP)+             ;RESTORE STACK POINTER
007156  005011                          CLR     @R1                     ;TURN OFF KT11-D
007160  005077  171632                  CLR     @TCSR
007164  012737  000066  000064  ODDAD:  MOV     #66,@#64                ;RESTORE TTY VECTOR RETURN TO CAUSE
007172  005037  000066                  CLR     @#66                    ;A HALT ON A FALSE INTERRUPT
007176  012737  000162  000160          MOV     #162,@#160
007204  005037  000162                  CLR     @#162
007210  005037  177776                  CLR     @#PS
007214  012737  007356  000104          MOV     #NG35E,@#104            ;SETUP INTERNAL TRAP SUCCESS RETURN
007222  005037  000106                  CLR     @#106
007226  012737  007366  000004          MOV     #OK35E,@#4              ;SETUP INTERNAL TRAP FAILURE RETURN
007234  005037  000006                  CLR     @#6
007240  012737  140000  177776          MOV     #140000,@#PS            ;SET MODE TO USER
007246  005277  171552                  INC     @SR0                    ;TURN ON KT11-D
007252  000000                          .
007254  000000                          .
007256  000000                          .
007260  000000                          .
007262  000000                          .
```

```
007264  000000                          0
007266  000000                          0
007270  000000                          0
007272  000000                          0
007274  000000                          0
007276  000000                          0
007300  000000                          0
007302  000000                          0
007304  000000                          0
007306  000000                          0
007310  000000                          0
007312  000000                          0
007314  000000                          0
007316  000000                          0
007320  000000                          0
007322  000000                          0
007324  000000                          0
007326  000000                          0
007330  000000                          0
007332  000000                          0
007334  000000                          0
007336  000000                          0
007340  000000                          0
007342  000000                          0
007344  000000                          0
007346  000000                          0
007350  000000                          0
007352  005737  000001                    TST     @#1                     ;ODD ADDRESS REFERENCE - AN "INTERNAL
                                                                        ;TRAP" SHOULD OCCUR
007356  022626                  NG35E1: CMP     (SP)+,(SP)+             ;RESTORE STACK POINTER
007360  005011                          CLR     @R1                     ;TURN OFF KT11-D
007362  104006                          HLT                             ;ODD ADDRESS TRAP DIDN'T TAKE
                                                                        ;VECTOR FROM KERNEL
007364  000402                          BR      END35
007366  022626                  OK35E1: CMP     (SP)+,(SP)+             ;RESTORE STACK POINTER
007370  005011                          CLR     @R1                     ;TURN OFF KT11-D
007372  012737  000006  000004  END351: MOV     #6,@#4                  ;RESTORE TRAP CATCHER
007400  012737  000106  000104          MOV     #106,@#104



                                        ;SHOW THAT THE ABORT LOGIC "LOCKS"  SR0, AND SR2 AFTER A NR
                                        ;ABORT UNTIL THE CORRESPONDING ABORT BIT IS CLEARED IN SR0, WHEN
                                        ;THEY RESUME TRACKING, A NR ERROR SHOULD STILL ABORT TO 250 EVEN
                                        ;WHEN BIT 15 (SR0) IS ALREADY SET
007406  104400                  TEST31: SCOPE
007410  012706  001000                  MOV     #KSTACK,SP              ;INITIALIZE KERNEL STACK POINTER
007414  004767  004420                  JSR     PC,ORDER                ;CHECK TEST SEQUENCE + INIT SR0
007420  000031                          31                              ;TEST NUMBER
007422  104006                          HLT                             ;TEST EXECUTED OUT OF SEQUENCE
007424  004767  005364                  JSR     X7,CLRALL               ;CLEAR ALL KT11-D REGISTERS
007430  004767  006470                  JSR     PC,KERN7                ;MAP KERNEL PAR/PDR 7 TO EXT BANK
007434  012777  077406  171432          MOV     #77406,@KPDR0           ;MAP KERNEL 0 RW,RK,BANK0
007442  012777  077400  171426          MOV     #77400,@KPDR1           ;MAP KERNEL 1 NR,4 K,BANK0
007450  012777  007504  171506          MOV     #INT36,@KTVEC           ;SETUP RETURN VECTOR
```

```
007456  005077  171504                  CLR     @KTSTA
007462  005277  171336                  INC     @SR0                    ;TURN ON KT11-D
007466  013737  037776  037776  ADR36:  MOV     @#37776,@#37776         ;REFERENCE KERNEL 1 - 1ST ABORT
007474  005077  171324                  CLR     @SR0                    ;TURN OFF KT11-D
007500  104006                          HLT                             ;REFERENCE TO KERNEL 1
007502  000510                          BR      DONE36                  ;DIDN'T ABORT
007504  042777  000001  171312  INT36:  BIC     #1,@SR0                 ;TURN OFF KT11-D
007512  022777  100002  171304          CMP     #100002,@SR0            ;CHECK SR0
007520  001401                          BEQ     .+4
007522  104006                          HLT                             ;SR0 INCORRECT AFTER NR ABORT
007524  012777  007560  171432          MOV     #INT36A,@KTVEC          ;SETUP NEW RETURN VECTOR
007532  022626                          CMP     (R6)+,(R6)+             ;RESTORE STACK POINTER
007534  012702  037776                  MOV     #37776,R2               ;SETUP R2 TO REFERENCE KERNEL 1
007540  052777  000001  171256          BIS     #1,@SR0                 ;TURN ON KT11-D
007546  012242                          MOV     (R2)+,-(R2)             ;REFERENCE KERNEL 1 -2ND ABORT
007550  005077  171250  ADR36A:  CLR     @SR0            ;TURN OFF KT11-D
007554  104006                          HLT
007556  000462                          BR      DONE36                  ;2ND REFERENCE TO KERNEL 1
007560  042777  000001  171236  INT36A: BIC     #1,@SR0         ;TURN OFF KT11-D
007566  022777  100002  171230          CMP     #100002,@SR0            ;DIDN'T ABORT
007574  001401                          BEQ     .+4                     ;CHECK SR0
007576  104006                          HLT
007600  022777  007466  171224          CMP     #ADR36,@SR2             ;SR0 INCORRECT AFTER 2ND NR ABORT
007606  001401                          BEQ     .+4                     ;CHECK SR2
007610  104006                          HLT
007612  021627  007550                  CMP     (R6),#ADR36A            ;SR2 DOESN'T CONTAIN VALUE FROM 1ST ABORT
007616  001401                          BEQ     .+4                     ;CHECK ADDRESS PUSHED ON STACK
007620  104006                          HLT
007622  022626                          CMP     (R6)+,(R6)+             ;INCORRECT ADDRESS ON STACK
007624  012777  007660  171332          MOV     #INT36B,@KTVEC          ;RESTORE STACK POINTER
007632  005077  171166                  CLR     @SR0                    ;CHANGE RETURN ADDRESS
                                                                        ;CLEAR NR ERROR BIT-SHOULD
                                                                        ;"UNLOCK" ERROR TRACKING
007636  012702  037776                  MOV     #37776,R2               ;SETUP R2 TO REFERENCE KERNEL 1
007642  005277  171156                  INC     @SR0                    ;TURN ON KT11-D
007646  012242          ADR36B: MOV     (R2)+,-(R2)     ;3RD NR REFERENCE, ERROR BIT WAS CLEARED
007650  005077  171150  ADR36C: CLR     @SR0            ;TURN OFF KT11-D
007654  104006                          HLT
007656  000422                          BR      DONE36                  ;3RD REFERENCE TO KERNEL 1
007660  042777  000001  171136  INT36B: BIC     #1,@SR0         ;TURN OFF KT11-D
007666  022777  100002  171130          CMP     #100002,@SR0            ;DIDN'T ABORT
007674  001401                          BEQ     .+4                     ;CHECK SR0
007676  104006                          HLT
007700  022777  007646  171124          CMP     #ADR36B,@SR2            ;SR0 INCORRECT
007706  001401                          BEQ     .+4                     ;CHECK SR2
007710  104006                          HLT

007712  022716  007650                  CMP     #ADR36C,(SP)            ;SR2 INCORRECT - SHOULD CONTAIN
007716  001401                          BEQ     .+4                     ;LAST FETCH ADDRESS BEFORE ABORT
007720  104006                          HLT                             ;CHECK STACK
007722  022626                          CMP     (R6)+,(R6)+             ;PC ON STACK INCORRECT
007724  005077  171074  DONE36: CLR     @#SR0                   ;RESTORE STACK POINTER
007730  005077  171232                  CLR     @KTSTA                  ;CLEAR ERROR BIT
007734  016777  171226  171222          MOV     KTSTA,@#C               ;CHANGE TRAP RETURN TO CAUSE A HALT
                                                                        ;ON A FALSE INTERRUPT
```

```
                                        ;SHOW THAT THE ABORT LOGIC "LOCKS" SR0, AND SR2 AFTER A PL
                                        ;ABORT UNTIL THE CORRESPONDING ABORT BIT IS CLEARED IN SR0, WHEN
                                        ;THEY RESUME TRACKING, A PL ERROR SHOULD STILL ABORT TO 250 EVEN
                                        ;WHEN BIT 14 (SR0) IS ALREADY SET
007742  104400                  TEST32: SCOPE
007744  012706  001000                  MOV     #KSTACK,SP              ;INITIALIZE KERNEL STACK POINTER
007750  004767  006064                  JSR     PC,ORDER                ;CHECK TEST SEQUENCE & INIT SR0
007754  000032                          32                              ;TEST NUMBER
007756  104006                          HLT                             ;TEST EXECUTED OUT OF SEQUENCE
007760  004767  005030                  JSR     %7,CLRALL               ;CLEAR ALL KT11-D REGISTERS
007764  004767  006134                  JSR     PC,KERN7                ;MAP KERNEL PAR/PDR 7 TO EXT BANK
007770  012777  077406  171076          MOV     #77406,@KPDR0           ;MAP KERNEL 0 RW,RK,BANK0
007776  012777  017406  171072          MOV     #17406,@KPDR1           ;MAP KERNEL 1 PL,1 K,BANK0
010004  012777  010040  171152          MOV     #INT37,@KTVEC           ;SETUP RETURN VECTOR
010012  005077  171150                  CLR     @KTSTA
010016  005277  171002                  INC     @SR0                    ;TURN ON KT11-D
010022  013737  037776  037776  ADR37:  MOV     @#37776,@#37776         ;REFERENCE KERNEL 1 - 1ST ABORT
010030  005077  170770                  CLR     @SR0                    ;TURN OFF KT11-D
010034  104006                          HLT                             ;REFERENCE TO KERNEL 1
010036  000510                          BR      DONE37                  ;DIDN'T ABORT
010040  042777  000001  170756  INT37:  BIC     #1,@SR0                 ;TURN OFF KT11-D
010046  022777  040002  170750          CMP     #40002,@SR0             ;CHECK SR0
010054  001401                          BEQ     .+4
010056  104006                          HLT                             ;SR0 INCORRECT AFTER PL ABORT
010060  012777  010114  171076          MOV     #INT37A,@KTVEC          ;SETUP NEW RETURN VECTOR
010066  022626                          CMP     (R6)+,(R6)+             ;RESTORE STACK POINTER
010070  012702  037776                  MOV     #37776,R2               ;SETUP R2 TO REFERENCE KERNEL 1
010074  052777  000001  170722          BIS     #1,@SR0                 ;TURN ON KT11-D
010102  012242                          MOV     (R2)+,-(R2)             ;REFERENCE KERNEL 1 -2ND ABORT
010104  005077  170714          ADR37A: CLR     @SR0            ;TURN OFF KT11-D
010110  104006                          HLT                             ;2ND REFERENCE TO KERNEL 1
010112  000462                          BR      DONE37                  ;DIDN'T ABORT
010114  042777  000001  170702  INT37A: BIC     #1,@SR0         ;TURN OFF KT11-D
010122  022777  040002  170674          CMP     #40002,@SR0             ;CHECK SR0
010130  001401                          BEQ     .+4
010132  104006                          HLT                             ;SR0 INCORRECT AFTER 2ND PL ABORT
010134  022777  010022  170670          CMP     #ADR37,@SR2             ;CHECK SR2
010142  001401                          BEQ     .+4
010144  104006                          HLT                             ;SR2 DOESN'T CONTAIN VALUE FROM 1ST ABORT
010146  021627  010104                  CMP     (R6),#ADR37A            ;CHECK ADDRESS PUSHED ON STACK
010152  001401                          BEQ     .+4
010154  104006                          HLT                             ;INCORRECT ADDRESS ON STACK
010156  022626                          CMP     (R6)+,(R6)+             ;RESTORE STACK POINTER
010160  012777  010214  170776          MOV     #INT37B,@KTVEC          ;CHANGE RETURN ADDRESS
010166  005077  170632                  CLR     @SR0                    ;CLEAR PL ERROR BIT-SHOULD
                                                                        ;"UNLOCK" ERROR TRACKING
010172  012702  037776                  MOV     #37776,R2               ;SETUP R2 TO REFERENCE KERNEL 1
010176  005277  170622                  INC     @SR0                    ;TURN ON KT11-D
010202  012242                  ADR37B: MOV     (R2)+,-(R2)     ;3RD PL REFERENCE, ERROR BIT WAS CLEARED
010204  005077  170614          ADR37C: CLR     @SR0            ;TURN OFF KT11-D
010210  104006                          HLT                             ;3RD REFERENCE TO KERNEL 1
010212  000422                          BR      DONE37                  ;DIDN'T ABORT
010214  042777  000001  170602  INT37B: BIC     #1,@SR0         ;TURN OFF KT11-D
010222  022777  040002  170574          CMP     #40002,@SR0             ;CHECK SR0
```

```
010230  001401                          BEQ     .+4
010232  104006                          HLT
010234  022777  010202  170570          CMP     #ADR37B,@SR2            ;CHECK SR2
010242  001401                          BEQ     .+4
010244  104006                          HLT                             ;SR2 INCORRECT - SHOULD CONTAIN
                                                                        ;LAST FETCH ADDRESS BEFORE ABORT
010246  022716  010204                  CMP     #ADR37C,(SP)            ;CHECK STACK
010252  001401                          BEQ     .+4
010254  104006                          HLT                             ;PC ON STACK INCORRECT
010256  022626                          CMP     (R6)+,(R6)+             ;RESTORE STACK POINTER
010260  005077  170540          DONE37: CLR     @SR0                    ;CLEAR ERROR BIT
010264  005077  170676                  CLR     @KTSTA                  ;CHANGE TRAP RETURN TO CAUSE A HALT
010270  016777  170672  170666          MOV     KTSTA,@KTVEC            ;ON A FALSE INTERRUPT

                                ;SHOW THAT THE ABORT LOGIC "LOCKS"  SR0, AND SR2 AFTER A ACC
                                ;ABORT UNTIL THE CORRESPONDING ABORT BIT IS CLEARED IN SR0, WHEN
                                ;THEY RESUME TRACKING, A ACC ERROR SHOULD STILL ABORT TO 250 EVEN
                                ;WHEN BIT 13 (SR0) IS ALREADY SET
010276  104400                  TEST33: SCOPE
010300  012706  001000                  MOV     #KSTACK,SP              ;INITIALIZE KERNEL STACK POINTER
010304  004767  005530                  JSR     PC,ORDER                ;CHECK TEST SEQUENCE + INIT SR0
010310  000033                          33                              ;TEST NUMBER
010312  104006                          HLT                             ;TEST EXECUTED OUT OF SEQUENCE
010314  004767  004474                  JSR     X7,CLRALL               ;CLEAR ALL KT11-D REGISTERS
010320  004767  005600                  JSR     PC,KERN7                ;MAP KERNEL PAR/PDR 7 TO EXT BANK
010324  012777  077406  170542          MOV     #77406,@KPDR0           ;MAP KERNEL 0 RW,RK,BANK0
010332  012777  077402  170536          MOV     #77402,@KPDR1           ;MAP KERNEL 1 ACC,4 K,BANK0
010340  012777  010374  170616          MOV     #INT40,@KTVEC           ;SETUP RETURN VECTOR
010346  005077  170614                  CLR     @KTSTA
010352  005277  170446                  INC     @SR0                    ;TURN ON KT11-D
010356  013737  037776  037776  ADR40:  MOV     @#37776,@#37776         ;REFERENCE KERNEL I - 1ST ABORT
010364  005077  170434                  CLR     @SR0                    ;TURN OFF KT11-D
010370  104006                          HLT                             ;REFERENCE TO KERNEL 1
010372  000510                          BR      DONE40                  ;DIDN'T ABORT
010374  042777  000001  170422  INT40:  BIC     #1,@SR0                 ;TURN OFF KT11-D
010402  022777  020002  170414          CMP     #20002,@SR0             ;CHECK SR0
010410  001401                          BEQ     .+4
010412  104006                          HLT
010414  012777  010450  170542          MOV     #INT40A,@KTVEC          ;SR0 INCORRECT AFTER ACC ABORT
010422  022626                          CMP     (R6)+,(R6)+             ;SETUP NEW RETURN VECTOR
010424  012702  037776                  MOV     #37776,R2               ;RESTORE STACK POINTER
010430  052777  000001  170366          BIS     #1,@SR0                 ;SETUP R2 TO REFERENCE KERNEL 1
010436  012242                          MOV     (R2)+,-(R2)             ;TURN ON KT11-D
010440  005077  170360  ADR40A: CLR     @SR0            ;REFERENCE KERNEL 1 -2ND ABORT
010444  104006                          HLT             ;TURN OFF KT11-D
010446  000462                          BR      DONE40                  ;2ND REFERENCE TO KERNEL 1
010450  042777  000001  170346  INT40A: BIC     #1,@SR0         ;DIDN'T ABORT
010456  022777  020002  170340          CMP     #20002,@SR0     ;TURN OFF KT11-D
010464  001401                          BEQ     .+4             ;CHECK SR0
010466  104006                          HLT
010470  022777  010356  170334          CMP     #ADR40,@SR2             ;SR0 INCORRECT AFTER 2ND ACC ABORT
010476  001401                          BEQ     .+4             ;CHECK SR2
010[illegible]  104006                  HLT
010[illegible]  021627  010440          CMP     (R6),#AD[illegible]A    ;SR2 DOESN'T CONTAIN VALUE FROM 1ST A[illegible]
                                                                        ;CHECK ADDRESS PUSHED ON STACK
```

```
010526  001401                          BEQ     .+4
010510  104026                          HLT
010512  022626                          CMP     (R6)+,(R6)+             ;INCORRECT ADDRESS ON STACK
010514  012777  010550  170442          MOV     #INT40B,@KTVEC          ;RESTORE STACK POINTER
010522  005077  170276                  CLR     @SR0                    ;CHANGE RETURN ADDRESS
                                                                        ;CLEAR ACC ERROR BIT-SHOULD
                                                                        ;"UNLOCK" ERROR TRACKING
010526  012702  037776                  MOV     #37776,R2               ;SETUP R2 TO REFERENCE KERNEL 1
010532  005277  170266                  INC     @SR0                    ;TURN ON KT11-D
010536  012242                  ADR40B: MOV     (R2)+,-(R2)     ;3RD ACC REFERENCE, ERROR BIT WAS CLEARED
010540  005077  170260          ADR40C: CLR     @SR0            ;TURN OFF KT11-D
010544  104006                          HLT
010546  000422                          BR      DONE40                  ;3RD REFERENCE TO KERNEL 1
010550  042777  000001  170246  INT40B: BIC     #1,@SR0                 ;DIDN'T ABORT
010556  022777  020002  170240          CMP     #20002,@SR0     ;TURN OFF KT11-D
010564  001401                          BEQ     .+4                     ;CHECK SR0
010566  104006                          HLT
010570  022777  010536  170234          CMP     #ADR40B,@SR2            ;SR0 INCORRECT
010576  001401                          BEQ     .+4                     ;CHECK SR2
010600  104006                          HLT
                                                                        ;SR2 INCORRECT - SHOULD CONTAIN
                                                                        ;LAST FETCH ADDRESS BEFORE ABORT
010602  022716  010540                  CMP     #ADR40C,(SP)            ;CHECK STACK
010606  001401                          BEQ     .+4
010610  104006                          HLT
010612  022626                          CMP     (R6)+,(R6)+             ;PC ON STACK INCORRECT
010614  005077  170204          DONE40: CLR     @SR0                    ;RESTORE STACK POINTER
010620  005077  170342                  CLR     @KTSTA                  ;CLEAR ERROR BIT
010624  016777  170336  170332          MOV     KTSTA,@KTVEC            ;CHANGE TRAP RETURN TO CAUSE A HALT
                                                                        ;ON A FALSE INTERRUPT

                                ;SHOW THAT INIT CLEARS SR0<13-15>
010632  104400                  TEST34: SCOPE
010634  012706  001000                  MOV     #KSTACK,SP              ;INITIALIZE KERNEL STACK POINTER
010640  004767  005174                  JSR     PC,ORDER                ;CHECK TEST SEQUENCE + INIT SR0
010644  000034                          34                              ;TEST NUMBER
010646  104006                          HLT                             ;TEST EXECUTED OUT OF SEQUENCE
010650  112777  000340  170150          MOVB    #340,@SR0H      ;SET SR0 BITS 13-15
010656  122777  000340  170142          CMPB    #340,@SR0H      ;MAKE SURE THEY SET CORRECTLY
010664  001401                          BEQ     .+4
010666  104006                          HLT                             ;SR0 INCORRECT (HIGH BYTE)
010670  000005                          RESET                           ;ISSUE INIT
010672  122777  000000  170126          CMPB    #0,@SR0H        ;CHECK SR0 HIGH BYTE
010700  001401                          BEQ     .+4
010702  104006                          HLT                             ;SR0 INCORRECT AFTER INIT
010704  012767  000010  004400          MOV     #10,ICOUNT      ;DROP ITERATION COUNT

                                ;SHOW THAT INIT CLEARS SR0 AFTER ABORT
010712  104400                  TEST35: SCOPE
010714  012706  001000                  MOV     #KSTACK,SP              ;INITIALIZE KERNEL STACK POINTER
010720  004767  005114                  JSR     PC,ORDER                ;CHECK TEST SEQUENCE + INIT SR0
010724  000035                          35                              ;TEST NUMBER
010726  104006                          HLT                             ;TEST EXECUTED OUT OF SEQUENCE
010730  004767  004104                  JSR     %7,RWALL        ;MAP ALL PAR/PDR PAIRS 4K,RW,BANK 0
010734  012777  000416  170132          MOV     #416,@KPDR0     ;MAP KERNEL 0 RW,4K LESS 1 PAGE
                                                                ;DOWN (100-17776 RW)
010742  004767  005156                  JSR     PC,KERN7                ;MAP KERNEL PAR/PDR 7 TO EXT BANK
```

```
010746  012777  077400  170122          MOV     #77400,@KPDR1   ;MAP KERNEL PAGE 1 NR
010754  012777  011010  170202          MOV     #RET2,@KTVEC    ;SETUP ABORT RETURN
010762  005077  170200                  CLR     @KTSTA
010766  012746  000020                  MOV     #20,-(SP)       ;SET T BIT IN STATUS ON STACK
010772  012746  011004                  MOV     #ADR2,-(SP)     ;SETUP ADDRESS ON STACK
010776  005277  170022                  INC     @SR0            ;TURN ON KT11-D
011002  000002                          RTI                     ;SHOULD TRACE TRAP IMMEDIATELY SINCE T-BIT
                                                                ;IS SET - SINCE T-BIT VECTOR IS OUTSIDE ALLOWED
                                                                ;PAGE LENGTH, SHOULD DO A MEMORY
                                                                ;MANAGEMENT ABORT

011004  000000          ADR2:   HALT                            ;NO PL ABORT OCCURRED
011006  000412                  BR      DONE2
011010  022777  040001  170006  RET2:   CMP     #40001,@SR0     ;CHECK SR0
011016  001401                  BEQ     .+4
011020  104006                  HLT                             ;SR0 INCORRECT - SHOULD SHOW
                                                                ;REFERENCE TO KERNEL 0
                                                                ;AND PL ABORT SHOULD BE SET

011022  000005                  RESET                           ;ISSUE INIT - SHOULD CLEAR SR0
011024  005777  167774          TST     @SR0            ;CHECK SR0
011030  001401                  BEQ     .+4
011032  104006                  HLT                             ;SR0 INCORRECT AFTER INIT
011034  005077  167764  DONE2:  CLR     @SR0            ;REINITIALIZE SR0
011040  016777  170122  170116          MOV     KTSTA,@KTVEC
011046  012737  000016  000014          MOV     #16,@#14        ;RESTORE T-BIT TRAP CATCHER

                                ;SHOW THAT INIT CLEARS SR0(0-3,5-6)
                                ;REFERENCE NR USER PAGE 7 TO SET ALL BITS(0-6)
                                ;THEN ISSUE INIT
011054  104400          TEST36: SCOPE
011056  012706  001000          MOV     #KSTACK,SP              ;INITIALIZE KERNEL STACK POINTER
011062  004767  004752          JSR     PC,ORDER                ;CHECK TEST SEQUENCE + INIT SR0
011066  000036                  36                              ;TEST NUMBER
011070  104006                  HLT                             ;TEST EXECUTED OUT OF SEQUENCE
011072  004767  003742          JSR     X7,RWALL        ;MAP ALL PAR/PDR PAIRS INITIALLY RW,4K,
                                                        ;BANK 0
011076  012777  077400  167746          MOV     #77400,@UPDR7   ;MAKE USER 7 NR
011104  004767  005014          JSR     PC,KERN7                ;MAP KERNEL PAR/PDR 7 TO EXT BANK
011110  012777  011146  170046          MOV     #RET3,@KTVEC    ;SETUP ABORT RETURN
011116  005077  170044          CLR     @KTSTA
011122  012737  140000  177776          MOV     #140000,@#PS    ;SET MODE TO USER
011130  012706  000400          MOV     #USTACK,R6      ;SETUP USER STACK IN CASE NEEDED
011134  005277  167664          INC     @SR0            ;TURN ON KT11-D
011140  005737  160000          TST     @#160000        ;REFERENCE PAGE 7
011144  000777                  BR      .               ;NO ABORT ON NR REFERENCE
011146  022777  100157  167650  RET3:   CMP     #100157,@SR0    ;CHECK SR0
011154  001401                  BEQ     .+4
011156  104006                  HLT                             ;SR0 INCORRECT - SHOULD HAVE TRACKED
                                                                ;NR REFERENCE TO USER 7
011160  000005                  RESET                           ;ISSUE INIT
011162  005777  167636          TST     @SR0            ;CHECK SR0
011166  001401                  BEQ     .+4
011170  104006                  HLT                             ;SR0 INCORRECT AFTER INIT
011172  005077  167626          CLR     @SR0
011176  012767  000010  004106          MOV     #10,ICOU        ;DROP ITERATION COUNT
```

```
011204  016777  167756  167752          MOV     KTSTA,@KTVEC

                                ;SHOW THAT BYTE ADDRESSING OF SR0 WORKS
011212  104400                  TEST37: SCOPE
011214  012706  001000                  MOV     #KSTACK,SP              ;INITIALIZE KERNEL STACK POINTER
011220  004767  004614                  JSR     PC,ORDER                ;CHECK TEST SEQUENCE + INIT SR0
011224  000037                          37                              ;TEST NUMBER
011226  104006                          HLT                             ;TEST EXECUTED OUT OF SEQUENCE
011230  004767  003604                  JSR     %7,RWALL        ;MAP ALL PAR/PDR PAIRS RW,4K,BANK 0
011234  004767  004664                  JSR     PC,KERN7                ;MAP KERNEL PAR/PDR 7 TO EXT BANK
011240  012777  160001  167556          MOV     #160001,@SR0    ;TURN ON KT11-D AND SET ERROR FLAGS
011246  105077  167552                  CLRB    @SR0            ;DATOB (LOW) TO SR0
011252  032777  160000  167544          BIT     #160000,@SR0    ;CHECK SR0
011260  001001                          BNE     .+4
011262  104006                          HLT                             ;SR0 INCORRECT AFTER DATOB
011264  012777  160001  167532          MOV     #160001,@SR0
011272  105077  167530                  CLRB    @SR0H           ;DATOB (HIGH) TO SR0
011276  022777  000017  167520          CMP     #17,@SR0                ;CHECK SR0
011304  001401                          BEQ     .+4
011306  104006                          HLT                             ;SR0 INCORRECT AFTER DATOB
011310  005077  167510                  CLR     @SR0

                                ;SHOW THAT SR0 <1-3> TRACK PAGE REFERENCED IF
                                ;KT11-D IS ON AND REFERENCE IS NOT TO A KT11-D REGISTER
                                ;SHOW THAT EACH VALUE IS CORRECTLY "LOCKED" IN SR0 AFTER AN ABORT
011314  104400                  TEST40: SCOPE
011316  012706  001000                  MOV     #KSTACK,SP              ;INITIALIZE KERNEL STACK POINTER
011322  004767  004512                  JSR     PC,ORDER                ;CHECK TEST SEQUENCE + INIT SR0
011326  000040                          40                              ;TEST NUMBER
011330  104006                          HLT                             ;TEST EXECUTED OUT OF SEQUENCE
011332  004767  003502                  JSR     %7,RWALL
011336  004767  004562                  JSR     PC,KERN7                ;MAP KERNEL PAR/PDR 7 TO EXT BANK
011342  012777  011420  167614          MOV     #RET5,@KTVEC
011350  005077  167612                  CLR     @KTSTA
011354  016701  167454                  MOV     UPDR0,R1
011360  005002                          CLR     R2
011362  012703  100141                  MOV     #100141,R3
011366  012704  000010                  MOV     #10,R4
011372  012711  077400          LOOP5:  MOV     #77400,@R1              ;MAKE USER NR
011376  012737  140000  177776          MOV     #140000,@#PS    ;ENTER USER MODE
011404  005277  167414                  INC     @SR0
011410  005712                          TST     @R2
011412  000777                          BR      .               ;REFERENCE TO NR PAGE DIDN'T ABORT
011414  000005                          RESET                   ;AFTER ERROR, TURN OFF KT11-D
011416  000423                          BR      DONE5
011420  017705  167400          RET5:   MOV     @SR0,R5         ;SAVE CONTENTS OF SR0
011424  005077  167374                  CLR     @SR0            ;TURN OFF KT11-D
011430  020503                          CMP     R5,R3           ;CHECK SAVED CONTENTS OF SR0
011432  001401                          BEQ     .+4
011434  104006                          HLT                             ;SR0 INCORRECT
011436  020167  167372                  CMP     R1,UPDR0        ;IS USER 0 UNDER TEST
011442  001002                          BNE     LOP5A           ;NO, CONTINUE
011444  012711  077406                  MOV     #77406,@R1      ;MAKE USER 0 RESIDENT
011450  022626                  LOP5A:  CMP     (R6)+,(R6)+
```

```
011452  005721                          TST     (R1)+
011454  062703  000002                  ADD     #2,R3
011460  062702  020000                  ADD     #20000,R2
011464  077436                          SOB     R4,LOOP5
011466  016777  167474  167470  DONE5:  MOV     KTSTA,@KTVEC
011474  005077  167466                  CLR     @KTSTA

                                ;SHOW THAT SR0 <5:6> TRACK PAGE REFERENCED (MODE) IF
                                ;KT11:D IS ON AND THE REFERENCE IS NOT TO A KT11:D REGISTER
                                ;SHOW THAT EACH VALUE IS CORRECTLY "LOCKED" IN SR0 AFTER AN ABORT
011500  104400                  TEST41: SCOPE
011502  012706  001000                  MOV     #KSTACK,SP              ;INITIALIZE KERNEL STACK POINTER
011506  004767  004326                  JSR     PC,ORDER                ;CHECK TEST SEQUENCE + INIT SR0
011512  000041                          41                              ;TEST NUMBER
011514  104006                          HLT                             ;TEST EXECUTED OUT OF SEQUENCE
011516  004767  003316                  JSR     %7,RWALL        ;MAP ALL PAGES RW,4K, BANK 0
011522  004767  004376                  JSR     PC,KERN7                ;MAP KERNEL PAR/PDR 7 TO EXT BANK
011526  012777  077400  167342          MOV     #77400,@KPDR1   ;SETUP PAGE 1 IN EACH MODE TO BE NR
011534  012777  077400  167274          MOV     #77400,@UPDR1
011542  012777  011570  167414          MOV     #RET7A,@KTVEC   ;SETUP ABORT RETURN
011550  005277  167250                  INC     @SR0            ;TURN ON KT11:D
011554  005737  020000                  TST     @#20000         ;REFERENCE PAGE 1 (NR)
011560  005077  167240                  CLR     @SR0            ;TURN OFF KT11:D
011564  104006                          HLT                     ;NR REFERENCE DIDN'T ABORT
011566  000436                          BR      DONE7
011570  017701  167230          RE7A:   MOV     @SR0,R1         ;SAVE SR0 CONTENTS IN R1
011574  005077  167224                  CLR     @SR0            ;TURN OFF KT11:D
011600  022701  100003                  CMP     #100003,R1      ;CHECK SAVED CONTENTS OF SR0
011604  001401                          BEQ     .+4
011606  104006                          HLT                     ;SR0 INCORRECT SHOULD SHOW NR ERR, KERNEL PAGE 1
011610  012777  011644  167346          MOV     #RET7C,@KTVEC   ;SETUP NEXT ABORT RETURN
011616  012737  140000  177776          MOV     #140000,@#PS    ;CHANGE MODE TO USER
011624  005277  167174                  INC     @SR0            ;TURN ON KT11:D
011630  005737  020000                  TST     @#20000         ;REFERENCE USER PAGE 1 (NR)
011634  005077  167164                  CLR     @SR0            ;TURN OFF KT11:D
011640  104006                          HLT                     ;NR REFERENCE DIDN'T ABORT
011642  000410                          BR      DONE7
011644  017701  167154          RE7C:   MOV     @SR0,R1         ;SAVE CONTENTS OF SR0
011650  005077  167150                  CLR     @SR0            ;TURN OFF KT11:D
011654  022701  100143                  CMP     #100143,R1      ;CHECK SAVED CONTENTS OF SR0
011660  001401                          BEQ     .+4
011662  104006                          HLT                     ;SR0 INCORRECT - SHOULD SHOW NR
                                                                ;ERROR, USER PAGE 1
011664  016777  167276  167272  DONE7:  MOV     KTSTA,@KTVEC    ;RESTORE TRAP CATCHER

                                ;SHOW THAT SR0 <1:3,5:6> DOESN'T TRACK IF KT11:D IS OFF BUT DOES IF REFERENCE IS TO
                                ;AN INTERNAL (KT11:D) REGISTER
011672  104400                  TEST42: SCOPE
011674  012706  001000                  MOV     #KSTACK,SP              ;INITIALIZE KERNEL STACK POINTER
011700  004767  004134                  JSR     PC,ORDER                ;CHECK TEST SEQUENCE + INIT SR0
011704  000042                          42                              ;TEST NUMBER
011706  104006                          HLT                             ;TEST EXECUTED OUT OF SEQUENCE
011[illegible]  004767  003124          JSR     %7,RWAL[illegible]      ;SET ALL PAR/PDR PAIRS RW, 4K, BANK 0
011[illegible]  012777  007600  167150  MOV     #7600,@U[illegible]7    ;MAP USER 7 TO THE EXT, BANK
```

```
011722  012737  140000  177776          MOV     #140000,@#PS    ;SET MODE TO USER
011730  005277  167070                  INC     @SR0            ;TURN ON KT11-D
011734  042777  000001  167062          BIC     #1,@SR0         ;TURN OFF KT11-D
011742  005037  177776                  CLR     @#PS            ;CHANGE TO KERNEL MODE
011746  022777  000156  167050          CMP     #156,@SR0       ;CHECK SR0
011754  001401                          BEQ     .+4
011756  104006                          HLT                     ;SR0 INCORRECT - SHOULD SHOW REFERENCE
                                                                ;TO USER 7
                                                                ;IF IT SHOWS USER 0
                                                                ;IT DID NOT TRACK THE INTERNAL REFERENCE
                                                                ;IF IT SHOWS KERNEL 0, IT IS
                                                                ;TRACKING WITH KT11-D OFF
011760  005077  167040                  CLR     @SR0

                                ;SHOW THAT IF AN INSTRUCTION IS COMPLETED BEFORE A MEMORY MANAGEMENT FAULT
                                ;OCCURS, SR2 WILL CONTAIN THE ADDRESS OFLAST FETCH BEFORE ABORT
                                ;TO TEST THIS, TRACE TRAP IS USED.  THE VECTOR IS MADE NON-RESIDENT BY MAKING
                                ;KERNEL PAGE 0 MAPPED DOWN FROM 17776 TO 100, THUS THE MEMORY MANAGEMENT
                                ;VECTOR IS RESIDENT WHILE THE TRACE TRAP VECTOR IS OUTSIDE THE ALLOWED
                                ;PAGE LENGTH.
011764  104400                  TEST43: SCOPE
011766  012706  001000                  MOV     #KSTACK,SP              ;INITIALIZE KERNEL STACK POINTER
011772  004767  004042                  JSR     PC,ORDER                ;CHECK TEST SEQUENCE + INIT SR0
011776  000043                          43                              ;TEST NUMBER
012000  104006                          HLT                             ;TEST EXECUTED OUT OF SEQUENCE
012002  004767  003032                  JSR     %7,RWALL        ;INITIALIZE ALL PAGES RW,4K,BANK 0
012006  012777  000416  167060          MOV     #416,@KPDR0     ;MAP KERNEL TO EXCLUDE
                                                                ;LOCATIONS 0 TO 77
012014  004767  004104                  JSR     PC,KERN7                ;MAP KERNEL PAR/PDR 7 TO EXT BANK
012020  012777  012056  167136          MOV     #RET11,@KTVEC   ;SETUP MEMORY MANAGEMENT ABORT RETURN
012026  005077  167134                  CLR     @KTSTA
012032  012746  000020                  MOV     #20,-(SP)       ;PREPARE STACK TO TURN ON T-BIT
012036  012746  012044                  MOV     #.+6,-(SP)
012042  000006                          RTT                     ;SET T-BIT VIA RTT
012044  012777  000001  166752  ADR11:  MOV     #1,@SR0         ;TURN ON KT11-D - SHOULD
                                                                ;ATTEMPT TO TRACE TRAP AT END OF
                                                                ;INSTRUCTION - SHOULD GET A PAGE
                                                                ;LENGTH ERROR ON THAT ATTEMPT
012052  000000                          HALT                    ;NO PAGE LENGTH ERROR ON TRACE TRAP
012054  000415                          BR      CONT11
012056  042777  000001  166740  RET11:  BIC     #1,@SR0         ;TURN OFF KT11-D
012064  022777  040000  166732          CMP     #40000,@SR0     ;CK SR0
012072  001401                          BEQ     .+4
012074  104006                          HLT                     ;SR0 INCORRECT - PL FAULT,KERNEL 0 REFERENCE COMPLETED
012076  022777  012044  166726          CMP     #ADR11,@SR2     ;CK SR2
012104  001401                          BEQ     .+4
012106  104006                          HLT                     ;SR2 INCORRECT - SHOULD CONTAIN
                                                                ;ADDRESS OF  LAST FETCH BEFORE ABORT
012110  005077  166710          CONT11: CLR     @SR0            ;REINITIALIZE SR0
012114  016777  167046  167042          MOV     KTSTA,@KTVEC    ;RESTORE TRAP CATCHER

                                ;SHOW THAT HAVING THE ABORT ERROR
                                ;BITS SET WILL NOT PREVENT A MEMORY MANAGEMENT TRAP
012122  104400                  TEST44: SCOPE
```

```
012124  012706  001000                  MOV     #KSTACK,SP              ;INITIALIZE KERNEL STACK POINTER
012130  004767  003704                  JSR     PC,ORDER                ;CHECK TEST SEQUENCE + INIT SR0
012134  000044                          44                              ;TEST NUMBER
012136  104006                          HLT                             ;TEST EXECUTED OUT OF SEQUENCE
012140  004767  002674                  JSR     %7,RWALL        ;INITIALIZE ALL PAR/PDR PAIRS TO RW,4K, BANK 0
012144  012777  077402  166726          MOV     #77402,@KPDR2   ;SET KERNEL PAR/PDR PAIR 2 RRO,4K
012152  004767  003746                  JSR     PC,KERN7                ;MAP KERNEL PAR/PDR 7 TO EXT BANK
012156  012777  012220  167000          MOV     #RET13A,@KTVEC  ;SETUP MEMORY MANAGEMENT ABORT RETURN
012164  005077  166776                  CLR     @KTSTA
012170  005277  166630                  INC     @SR0            ;TURN ON KT11-D
012174  012777  160001  166622          MOV     #160001,@SR0    ;SET ABORT ERROR BITS
012202  013737  007000  047000          MOV     @#7000,@#47000  ;WRITE KERNEL PAR/PDR PAIR 2 (RRO)-SHOULD TRAP
012210  005077  166610                  CLR     @SR0
012214  104006                          HLT                             ;NO TRAP OCCURRED
012216  000416                          BR      DONE13
012220  022626                  RET13A: CMP     (SP)+,(SP)+             ;RESTORE THE STACK POINTER
012222  017701  166576                  MOV     @SR0,R1                 ;SAVE CONTENTS OF SR0
012226  005077  166572                  CLR     @SR0                    ;TURN OFF KT11-D
012232  022701  160017                  CMP     #160017,R1
012236  001401                          BEQ     .+4
012240  104006                          HLT                             ;SAVED CONTENTS OF SR0 INCORRECT
012242  022777  077402  166630          CMP     #77402,@KPDR2   ;CHECK THE PDR CORRESPONDING TO THE TRAP REFERENCE
012250  001401                          BEQ     .+4
012252  104006                          HLT                             ;THE PDR CORRESPONDING TO THE TRAP REFERENCE IS INCORREC
012254  016777  166706  166702  DONE13: MOV     KTSTA,@KTVEC    ;RESTORE MEMORY MANAGEMENT TRAP RETURN
                                                                        ;TO CAUSE A HALT ON A FALSE TRAP OR ABORT
012262  005077  166536                  CLR     @SR0            ;REINITIALIZE SR0

                                ;SHOW THAT MEMORY MANAGEMENT WILL NOT TRAP ON AN INTERNAL REFERENCE
012266  104400                  TEST45: SCOPE
012270  012706  001000                  MOV     #KSTACK,SP              ;INITIALIZE KERNEL STACK POINTER
012274  004767  003540                  JSR     PC,ORDER                ;CHECK TEST SEQUENCE + INIT SR0
012300  000045                          45                              ;TEST NUMBER
012302  104006                          HLT                             ;TEST EXECUTED OUT OF SEQUENCE
012304  004767  002530                  JSR     %7,RWALL        ;MAP ALL PAR/PDR PAIRS 4K, RW, BANK 0
012310  004767  003610                  JSR     PC,KERN7                ;MAP KERNEL PAR/PDR 7 TO EXT BANK
012314  012777  012356  166642          MOV     #RET16,@KTVEC   ;SETUP TRAP RETURN IN CASE
012322  005077  166640                  CLR     @KTSTA
012326  005277  166472                  INC     @SR0            ;TURN ON KT11-D
012332  005777  166466                  TST     @SR0            ;TRAP REFERENCE TO A KT11-D REGISTER
012336  005077  166462                  CLR     @SR0
012342  022777  077406  166542          CMP     #77406,@KPDR7
012350  001401                          BEQ     .+4
012352  104006                          HLT
012354  000404                          BR      DONE16
012356  042777  000001  166440  RET16:  BIC     #1,@SR0
012364  104006                          HLT
012366  005077  166432          DONE16: CLR     @SR0
012372  016777  166570  166564          MOV     KTSTA,@KTVEC

                                ;TEST PAGE LENGTH ERROR CHECKING (EXPAND DOWN NOT SET)
                                ;KERNEL PAR/PDR PAIR1 IS USED WITH ALL PAGE LENGTH VALUES
                                ;SHOW THAT REFERENCES TO [illegible]TH BOUNDARIES OF THE ALLOWED AREA DON'T TRAP OR A
                                ;SHOW THAT A REFERENCE TO THE FIRST WORD BEYOND THE ALLOWABLE AREA DOES TRAP
```

```
012400  104400                  TEST46: SCOPE
012402  012706  001000                  MOV     #KSTACK,SP              ;INITIALIZE KERNEL STACK POINTER
012406  004767  003426                  JSR     PC,ORDER                ;CHECK TEST SEQUENCE & INIT SR0
012412  000046                          46                              ;TEST NUMBER
012414  104006                          HLT                             ;TEST EXECUTED OUT OF SEQUENCE
012416  004767  002416                  JSR     %7,RWALL        ;INITIALIZE ALL PAR/PDR PAIRS TO RW,4K, BANK  0
012422  004767  003476                  JSR     PC,KERN7                ;MAP KERNEL PAR/PDR 7 TO EXT BANK
012426  012702  000006                  MOV     #6,R2           ;R2 CONTAINS VALUE TO BE LOADED IN THE
                                                                ;PDR BEING CHECKED (INCLUDING PLF)
012432  012701  020076                  MOV     #20076,R1       ;R1 IS USED TO REFERENCE THE TOP ADDRESS
                                                                ;WITHIN THE ALLOWED AREA
012436  012777  012516  166520          MOV     #RET23A,@KTVEC  ;SETUP ABORT RETURN IN CASE REFERENCE
012444  005077  166516                  CLR     @KTSTA          ;WITHIN ALLOWED AREA ABORTS
012450  005277  166350          LOOP23: INC     @SR0            ;TURN ON KT11-D
012454  010277  166416                  MOV     R2,@KPDR1       ;SET KERNEL PAR/PDR PAIR 1 TO NEW PAGE LENGTH
012460  005727  020000                  TST     #20000          ;READ LOWER BOUNDARY-SHOULDN'T ABORT
012464  005711                          TST     @R1             ;READ UPPER ALLOWED BOUNDARY-SHOULDN'T
                                                                ;ABORT
012466  012777  012536  166470          MOV     #RET23B,@KTVEC  ;SETUP ABORT RETURN
012474  020127  037776                  CMP     R1,#37776       ;CHECK FOR DONE (TO AVOID REFERENCING
                                                                ;NEXT PAR/PDR PAIR)
012500  103041                          BHIS    DONE23          ;EXIT LOOP IF DONE
012502  005761  000002                  TST     2(R1)           ;REFERENCE OUTSIDE ALLOWED AREA -
                                                                ;SHOULD ABORT
012506  005077  166312                  CLR     @SR0            ;TURN KT11-D OFF
012512  104006                          HLT                     ;NO ABORT OCCURRED ON A REFERENCE
012514  000426                          BR      CONT23          ;OUTSIDE THE ALLOWED PAGE LENGTH
012516  042777  000001  166300  RET23A: BIC     #1,@SR0         ;TURN OFF KT11-D
012524  022626                  CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
012526  104006                          HLT                     ;REFERENCE WITHIN ALLOWED AREA
012530  005077  166270                  CLR     @SR0            ;CLEAR ERROR BITS
012534  000416                          BR      CONT23          ;CAUSED A TRAP OR ABORT
012536  022626                  RET23B: CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
012540  017703  166260                  MOV     @SR0,R3         ;SAVE CURRENT SR0
012544  005077  166254                  CLR     @SR0            ;TURN OFF KT11-D
012550  022703  040003                  CMP     #40003,R3       ;CK SAVED SR0
012554  001401                          BEQ     .+4
012556  104006                          HLT
                                                                ;CONTENTS OF SR0 INCORRECT AFTER
                                                                ;PAGE LENGTH ERROR ABORT
012560  022777  000002  166236          CMP     #2,@SR0         ;CHECK SR0 TO BE SURE PL BIT CLEARED
012566  001401                          BEQ     .+4
012570  104006                          HLT
                                                                ;SR0 INCORRECT AFTER CLEARING IT
                                                                ;ONLY KERNEL PAGE 1 SHOULD STILL BE SET
012572  062701  000100          CONT23: ADD     #100,R1         ;SETUP R1 TO REFERENCE BOUNDARY OF
                                                                ;NEXT PAGE
012576  062702  000400                  ADD     #400,R2         ;ADD 1 TO VALUE TO BE LOADED IN
                                                                ;PAGE LENGTH FIELD
012602  000722                          BR      LOOP23          ;CHECK NEXT PAGE LENGTH VALUE
012604  005077  166214          DONE23: CLR     @SR0            ;TURN OFF KT11-D
012610  016777  166352  166346          MOV     KTSTA,@KTVEC    ;RESTORE MEMORY MANAGEMENT ABORT RETURN
012616  005077  166344                  CLR     @KTSTA          ;TO CAUSE HALT ON A FALSE TRAP
                                                                ;OR ABORT

                        ;TEST PAGE LENGTH ERROR CHECKING (EXPAND DOWN SET)
```

```
                                ;KERNEL PAR/PDR PAIR1 IS TESTED WITH ALL VALUES OF PAGE LENGTH FIELD
                                ;SHOW THAT REFERENCES TO BOTH BOUNDARIES OF THE ALLOWED AREA DON'T TRAP OR ABORT
                                ;SHOW THAT A REFERENCE TO THE WORD IMMEDIATELY BELOW THE ALLOWED AREA DOES TRAP
012622  104400                  TEST47: SCOPE
012624  012706  001000                  MOV     #KSTACK,SP              ;INITIALIZE KERNEL STACK POINTER
012630  004767  003204                  JSR     PC,ORDER                ;CHECK TEST SEQUENCE + INIT SR0
012634  000047                          47                              ;TEST NUMBER
012636  104006                          HLT                             ;TEST EXECUTED OUT OF SEQUENCE
012640  004767  002174                  JSR     X7,RWALL        ;INITIALIZE ALL PAR/PDR PAIRS TO RW,4K, BANK 0
012644  004767  003254                  JSR     PC,KERN7                ;MAP KERNEL PAR/PDR 7 TO EXT BANK
012650  012702  077416                  MOV     #77416,R2       ;R2 CONTAINS VALUE TO BE LOADED IN THE
                                                                ;PDR BEING CHECKING (INCLUDING PLF)
012654  012701  037700                  MOV     #37700,R1       ;R1 IS USED TO REFERENCE THE LOWEST
                                                                ;ALLOWED ADDRESS IN THE PAGE
012660  012777  012740  166276          MOV     #RET24A,@KTVEC  ;SETUP ABORT RETURN IN CASE REFERENCE
012666  005077  166274                  CLR     @KTSTA          ;WITHIN ALLOWED AREA ABORTS
012672  005277  166126          LOOP24: INC     @SR0            ;TURN ON KT11-D
012676  010277  166174                  MOV     R2,@KPDR1       ;SET KERNEL PAR/PDR PAIR 1 TO NEW PAGE LENGTH
012702  005727  037776                  TST     #37776          ;REFERENCE UPPER ALLOWED BOUNDARY
012706  005711                          TST     @R1             ;REFERENCE LOWER ALLOWED BOUNDARY
                                                                ;- NEITHER REFERENCE SHOULD ABORT
012710  012777  012752  166246          MOV     #RET24B,@KTVEC  ;SETUP ABORT RETURN
012716  020127  020000                  CMP     R1,#20000       ;CHECK FOR DONE
012722  001436                          BEQ     DONE24          ;EXIT LOOP IF DONE
012724  005761  177776                  TST     -2(R1)          ;REFERENCE BELOW ALLOWED AREA -
                                                                ;SHOULD ABORT
012730  005077  166070                  CLR     @SR0            ;TURN KT11-D OFF
012734  104006                          HLT                     ;NO ABORT OCCURRED ON A REFERENCE
012736  000423                          BR      CONT24          ;OUTSIDE THE ALLOWED PAGE LENGTH
012740  005077  166060          RET24A: CLR     @SR0            ;TURN OFF KT11-D AND CLEAR
                                                                ;ERROR BITS
012744  022626                          CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
012746  104006                          HLT                     ;REFERENCE WITHIN ALLOWED AREA CAUSED
012750  000416                          BR      CONT24          ;A TRAP OR ABORT
012752  022626          RET24B: CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
012754  017703  166044                  MOV     @SR0,R3         ;SAVE CURRENT SR0
012760  005077  166040                  CLR     @SR0            ;TURN OFF KT11-D
012764  022703  040003                  CMP     #40003,R3       ;CK SAVED SR0
012770  001401                          BEQ     .+4
012772  104006                          HLT                     ;CONTENTS OF SR0 INCORRECT AFTER
                                                                ;PAGE LENGTH ERROR ABORT
012774  022777  000002  166022          CMP     #2,@SR0         ;CHECK SR0 TO BE SURE PL BIT CLEARED
013002  001401                          BEQ     .+4
013004  104006                          HLT                     ;SR0 INCORRECT AFTER CLEARING IT
013006  162701  000100          CONT24: SUB     #100,R1         ;SETUP R1 TO REFERENCE BOUNDARY
                                                                ;OF NEXT PAGE DOWN
013012  162702  000400                  SUB     #400,R2         ;INCREASE ALLOWED PAGE LENGTH
                                                                ;(DOWN) BY 1 PAGE
013016  000725                          BR      LOOP24          ;CHECK NEXT PAGE LENGTH VALUE
013020  005077  166000          DONE24: CLR     @SR0            ;TURN OFF KT11-D
013024  016777  166136  166132          MOV     KTSTA,@KTVEC    ;RESTORE MEMORY MANAGEMENT ABORT RETURN
013032  005077  166130                  CLR     @KTSTA          ;TO CAUSE A HALT ON A FALSE TRAP
                                                                ;OR ABORT
```

```
                                        ;TEST ALL COMBINATIONS OF VALUES FOR THE PAGE LENGTH COMPARATORS.
                                        ;USE KERNEL PAGE PAGE 1
013036  104400                  TEST50: SCOPE
013040  012706  001000                  MOV     #KSTACK,SP              ;INITIALIZE KERNEL STACK POINTER
013044  004767  002770                  JSR     PC,ORDER                ;CHECK TEST SEQUENCE & INIT SR0
013050  000050                          50                              ;TEST NUMBER
013052  104006                          HLT                             ;TEST EXECUTED OUT OF SEQUENCE
013054  012767  000020  002230          MOV     #20,ICOUNT      ;DROP ITERATION COUNT
013062  004767  001752                  JSR     %7,RWALL        ;INITIALIZE ALL PAGES RW, BANK 0
013066  004767  003032                  JSR     PC,KERN7                ;MAP KERNEL PAR/PDR 7 TO EXT BANK
013072  012777  013204  166064          MOV     #RET25,@KTVEC   ;SETUP ABORT RETURN
013100  005077  166062                  CLR     @KTSTA
013104  012701  000006                  MOV     #6,R1           ;R1 CONTAINS THE VALUE TO BE
                                                                ;LOADED INTO THE PDR
013110  012777  000001  165706          MOV     #1,@SR0         ;TURN ON KT11.D
013116  012703  020000          L25A:   MOV     #20000,R3       ;R3 CONTAINS VA USED
013122  010177  165750                  MOV     R1,@KPDR1       ;LOAD NEW PAGE LENGTH FIELD
013126  010102                  L25B:   MOV     R1,R2           ;R2 IS A COPY OF R1
013130  010304                          MOV     R3,R4           ;R4 IS A COPY OF R3
013132  042704  160000                  BIC     #160000,R4
013136  005713                          TST     (R3)            ;USE VA IN R3 TO REFERENCE PAGE 1
013140  000302                          SWAB    R2              ;NO TRAP.CHECK TO MAKE SURE
013142  042702  177400                  BIC     #177400,R2
013146  006304                          ASL     R4              ;VIRTUAL ADDRESS WAS WITHIN
013150  006304                          ASL     R4
013152  000304                          SWAB    R4
013154  020402                          CMP     R4,R2           ;ALLOWED PAGE LENGTH
013156  003401                          BLE     .+4
013160  104006                          HLT                             ;REFERENCE OUTSIDE ALLOWED PAGE LENGTH
                                                                ;DIDN'T ABORT
013162  062703  000100          C25:    ADD     #100,R3
013166  020327  037776                  CMP     R3,#37776
013172  003755                          BLE     L25B
013174  062701  000400                  ADD     #400,R1
013200  100346                          BPL     L25A
013202  000413                          BR      DONE25
013204  022626                  RET25:  CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
013206  000302                          SWAB    R2              ;CHECK TO MAKE SURE VIRTUAL
013210  042702  177400                  BIC     #177400,R2
013214  006304                          ASL     R4              ;ADDRESS WAS OUTSIDE ALLOWED
013216  006304                          ASL     R4
013220  000304                          SWAB    R4
013222  020402                          CMP     R4,R2           ;PAGE LENGTH
013224  003001                          BGT     .+4
013226  104006                          HLT                             ;REFERENCE WITHIN ALLOWED
013230  000754                          BR      C25             ;PAGE LENGTH ABORTED.R3 CONTAINS
                                                                ;VA USED, R1 CONTAINS VALUE
                                                                ;LOADED INTO THE PDR
013232  016777  165730  165724  DONE25: MOV     KTSTA,@KTVEC
013240  005077  165560                  CLR     @SR0

                                        ;SHOW THAT THE W BIT DOESN'T SET IF THE KT11.D IS OFF
013244  104400                  TEST51: SCOPE
013246  012706  001000                  MOV     #KSTACK,SP              ;INITIALIZE KERNEL STACK POINTER
```

```
013252  004767  002562                  JSR     PC,ORDER                ;CHECK TEST SEQUENCE & INIT SR0
013256  000051                          51                              ;TEST NUMBER
013260  104006                          HLT                             ;TEST EXECUTED OUT OF SEQUENCE
013262  012767  002000  002022          MOV     #2000,ICOUNT    ;RESTORE ITERATION COUNT
013270  004767  001520                  JSR     %7,CLRALL       ;CLEAR ALL KT11-D REGISTERS
013274  013737  010000  010000          MOV     @#10000,@#10000 ;WRITE BANK 0
013302  005777  165566                  TST     @KPDR0
013306  001401                          BEQ     .+4
013310  104006                          HLT                             ;W BIT SET OR ANOTHER BIT INCORRECT
                                                                        ;IN KERNEL 0 PDR

                                ;SHOW THAT THE W BIT IS CLEARED BY WRITING (VIA DATO) THE CORRESPONDING PAR
                                ;CHECK EACH PDR
013312  104400                  TEST52: SCOPE
013314  012706  001000                  MOV     #KSTACK,SP              ;INITIALIZE KERNEL STACK POINTER
013320  004767  002514                  JSR     PC,ORDER                ;CHECK TEST SEQUENCE & INIT SR0
013324  000052                          52                              ;TEST NUMBER
013326  104006                          HLT                             ;TEST EXECUTED OUT OF SEQUENCE
013330  004767  001504                  JSR     %7,RWALL
013334  004767  002564                  JSR     PC,KERN7                ;MAP KERNEL PAR/PDR 7 TO EXT BANK
013340  012777  007600  165524          MOV     #7600,@UPAR7    ;MAP USER 7 TO EXTERNAL BANK
013346  012737  140000  177776          MOV     #140000,@#PS    ;SET MODE TO USER
013354  012706  000400                  MOV     #USTACK,R6      ;SETUP USER STACK
013360  012700  001144                  MOV     #STATAB,R0      ;SET UP KT REG TABLE POINTER
013364  012001                  LOP27:  MOV     (R0)+,R1        ;R1 CONTAINS ADDRESS OF
                                                                ;ADDRESS OF CURRENT PDR
013366  012702  017776                  MOV     #17776,R2       ;R2 CONTAINS VIRTUAL ADDRESS TO
                                                                ;REFERENCE DESIRED PAGE
013372  012037  177776                  MOV     (R0)+,@#PS      ;SETUP STATUS FOR CURRENT MODE
013376  005277  165422          LOP27A: INC     @SR0            ;TURN ON KT11-D
013402  011212                          MOV     (R2),(R2)       ;WRITE
013404  005077  165414                  CLR     @SR0            ;TURN OFF KT11-D
013410  004767  000016                  JSR     %7,CKWBIT       ;TEST W BIT
013414  062702  020000                  ADD     #20000,R2       ;CHANGE VA TO REFERENCE NEXT PAGE
013420  103366                          BCC     LOP27A          ;LOOP UNTIL ALL PDR'S HAVE BEEN
                                                                ;CHECKED IN THE CURRENT MODE
013422  020027  001152                  CMP     R0,#STAEND
013426  002756                          BLT     LOP27
013430  000416                          BR      EXT27
013432  032771  000100  000000  CKWBIT: BIT     #100,@(R1)      ;CHECK W BIT
013440  001001                          BNE     .+4
013442  104006                          HLT                             ;W BIT DIDN'T SET IN PDR WHOSE
                                                                        ;ADDRESS IS POINTED TO BY R1
013444  005071  000020                  CLR     @20(R1)         ;CLEAR W BIT BY WRITING CORRESPONDING
                                                                        ;PAR VIA DATO
013450  032771  000100  000000          BIT     #100,@(R1)      ;CHECK W BIT
013456  001401                          BEQ     .+4
013460  104006                          HLT                             ;W BIT DIDN'T CLEAR IN PDR WHOSE
013462  005721                          TST     (R1)+           ;ADDRESS IS POINTED TO BY R1
013464  000207                          RTS     %7
013466                          EXT27:

                                ;SHOW THAT THE W BIT IS CLEARED BY A DATOB TO THE PDR
                                ;CHECK BOTH HIGH AND LOW DATOB'S, ON KERNEL 0
```

```
013466  104420                  TEST53: SCOPE
013470  012706  001000                  MOV     #KSTACK,SP              ;INITIALIZE KERNEL STACK POINTER
013474  004767  002340                  JSR     PC,ORDER                ;CHECK TEST SEQUENCE + INIT SR0
013500  000053                          53                              ;TEST NUMBER
013502  104006                          HLT                             ;TEST EXECUTED OUT OF SEQUENCE
013504  004767  001330                  JSR     %7,RWALL        ;MAP ALL PAR/PDR PAIRS 4K, RW, BANK 0
013510  004767  002410                  JSR     PC,KERN7                ;MAP KERNEL PAR/PDR 7 TO EXT BANK
013514  005277  165304                  INC     @SR0            ;TURN ON KT11-D
013520  013737  000000  000000          MOV     @#0,@#0         ;WRITE INTO PAGE 0
013526  005077  165272                  CLR     @SR0            ;TURN OFF KT11-D
013532  032777  000100  165334          BIT     #100,@KPDR0     ;CHECK W BIT
013540  001001                          BNE     .+4
013542  104006                          HLT                             ;W BIT NOT SET AFTER WRITING PAGE
013544  112777  000106  165322          MOVB    #106,@KPDR0     ;DATOB SHOULD CLEAR W BIT
013552  032777  000100  165314          BIT     #100,@KPDR0
013560  001401                          BEQ     .+4
013562  104006                          HLT                             ;W BIT DIDN'T CLEAR VIA DATOB (LOW)
                                                                        ;TO THE PDR
013564  005277  165234                  INC     @SR0            ;TURN ON KT11-D
013570  013737  017776  017776          MOV     @#17776,@#17776 ;WRITE INTO PAGE 0 AGAIN
013576  005077  165222                  CLR     @SR0            ;TURN OFF KT11-D
013602  032777  000100  165264          BIT     #100,@KPDR0     ;CHECK W BIT
013610  001001                          BNE     .+4
013612  104006                          HLT                             ;W BIT NOT SET AFTER WRITING PAGE
013614  016701  165254                  MOV     KPDR0,R1        ;SETUP R1 TO REFERENCE HIGH BYTE
013620  005201                          INC     R1                      ;OF KPDR0
013622  112711  000177                  MOVB    #177,@R1        ;DATOB TO HIGH BYTE OF KPDR0
013626  032777  000100  165240          BIT     #100,@KPDR0     ;CHECK W BIT
013634  001401                          BEQ     .+4
013636  104006                          HLT                             ;W BIT DIDN'T CLEAR VIA DATOB
                                                                        ;TO HIGH BYTE OF PDR
                                ;SHOW THAT THE W BIT IS CLEARED BY A DATOB TO THE PAR
                                ;CHECK BOTH HIGH AND LOW DATOB'S, ON KERNEL 0
013640  104400                  TEST54: SCOPE
013642  012706  001000                  MOV     #KSTACK,SP              ;INITIALIZE KERNEL STACK POINTER
013646  004767  002166                  JSR     PC,ORDER                ;CHECK TEST SEQUENCE + INIT SR0
013652  000054                          54                              ;TEST NUMBER
013654  104006                          HLT                             ;TEST EXECUTED OUT OF SEQUENCE
013656  004767  001156                  JSR     %7,RWALL        ;MAP ALL PAR/PDR PAIRS 4K, RW, BANK 0
013662  004767  002236                  JSR     PC,KERN7                ;MAP KERNEL PAR/PDR 7 TO EXT BANK
                                                                        ;EXTERNAL BANK
013666  005277  165132                  INC     @SR0            ;TURN ON KT11-D
013672  013737  000000  000000          MOV     @#0,@#0         ;WRITE INTO PAGE 0
013700  005077  165120                  CLR     @SR0            ;TURN OFF KT11-D
013704  032777  000100  165162          BIT     #100,@KPDR0     ;CHECK W BIT
013712  001001                          BNE     .+4
013714  104006                          HLT                             ;W BIT NOT SET AFTER WRITING PAGE
013716  112777  000000  165170          MOVB    #0,@KPAR0       ;DATOB TO THE PAR
013724  032777  000100  165142          BIT     #100,@KPDR0     ;CHECK W BIT
013732  001401                          BEQ     .+4
013734  104006                          HLT                             ;W BIT DIDN'T CLEAR VIA DATOB
                                                                        ;(LOW) TO THE PAR
013736  005277  165062                  INC     @SR0            ;TURN ON KT11-D
013742  013737  017776  017776          MOV     @#17776,@#17776 ;WRITE INTO PAGE 0 AGAIN
```

```
013750  005077  165050                  CLR     @SR0            ;TURN OFF KT11-D
013754  032777  000100  165112          BIT     #100,@KPDR0     ;CHECK W BIT
013762  001001                          BNE     .+4
013764  104006                          HLT                     ;W BIT NOT SET AFTER WRITING PAGE
013766  016701  165122                  MOV     KPAR0,R1        ;SETUP R1 TO REFERENCE HIGH BYTE
013772  005201                          INC     R1              ;OF KPAR0
013774  112711  000000                  MOVB    #0,@R1          ;DATOB TO HIGH BYTE OF KPAR0
014000  032777  000100  165066          BIT     #100,@KPDR0     ;CHECK W BIT
014006  001401                          BEQ     .+4
014010  104006                          HLT                     ;W BIT DIDN'T CLEAR VIA DATOB
                                                                ;TO HIGH BYTE OF PAR

                                ;SHOW THAT THE W BIT IS NOT CLEARED BY INIT
                                ;INITIALLY SET ALL THE W BITS, THEN DO A RESET AND CHECK THE W BITS
014012  104400                  TEST55: SCOPE
014014  012706  001000                  MOV     #KSTACK,SP              ;INITIALIZE KERNEL STACK POINTER
014020  004767  002014                  JSR     PC,ORDER                ;CHECK TEST SEQUENCE + INIT SR0
014024  000055                          55                              ;TEST NUMBER
014026  104006                          HLT                             ;TEST EXECUTED OUT OF SEQUENCE
014030  004767  001004                  JSR     %7,RWALL        ;INITIALIZE ALL PAGES RW,4K,BANK 0
014034  004767  002064                  JSR     PC,KERN7                ;MAP KERNEL PAR/PDR 7 TO EXT BANK
014040  012777  007600  165024          MOV     #7600,@UPAR7    ;MAP USER 7 TO THE EXTERNAL BANK
014046  012737  140000  177776          MOV     #140000,@#PS    ;SET MODE TO USER
014054  012706  000400                  MOV     #USTACK,R6      ;SETUP USER STACK
014060  012700  001144                  MOV     #STATAB,R0      ;R0 POINTS TO INFORMATION FOR
                                                                ;CURRENT MODE
014064  005720          LOOP32: TST     (R0)+                   ;MOVE POINTER
014066  012037  177776                  MOV     (R0)+,@#PS      ;SETUP MODE TO REFERENCE NEXT SET OF REGS
014072  012702  017776                  MOV     #17776,R2       ;SETUP R2 TO REFERENCE DESIRED PAGE
014076  005277  164722                  INC     @SR0
014102  011212          LOP32C: MOV     (R2),(R2)               ;WRITE IN
014104  062702  020000                  ADD     #20000,R2       ;CHANGE VA TO REFERENCE NEXT PAGE
014110  103374                          BCC     LOP32C          ;SET ALL W-BITS IN CURRENT MODE
014112  005077  164706                  CLR     @SR0            ;TURN OFF KT11-D
014116  020027  001152                  CMP     R0,#STAEND      ;CHECK FOR DONE SETTING THE W BITS
014122  002760                          BLT     LOOP32          ;IF NOT, LOOP TO DO NEXT MODE
014124  012701  001034                  MOV     #ADRTAB,R1      ;SETUP R1 TO REFERENCE ADDRESSES OF PDR'S OF PDR'S
014130  012702  000010          LOP32D: MOV     #10,R2          ;USE R2 AS COUNTER TO CHANGE ADDRESS
                                                                ;AT END OF EACH SET OF REGISTERS
014134  032771  000100  000000  LOP32E: BIT     #100,@(R1)      ;CHECK W BIT
014142  001001                          BNE     .+4
014144  104006                          HLT                     ;W BIT NOT SET IN PDR WHOSE
                                                                ;ADDRESS IS POINTED TO BY R1-
                                                                ;SHOULD HAVE BEEN SET WHEN
                                                                ;PAGE WAS WRITTEN INTO
014146  005721                          TST     (R1)+           ;MOVE POINTER
014150  077207                          SOB     R2,LOP32E       ;CHECK ALL PDR'S IN THIS SET
014152  062701  000020                  ADD     #20,R1          ;CHANGE R1 TO REFERENCE NEXT
                                                                ;SET OF PDR ADDRESSES
014156  020127  001132                  CMP     R1,#ADREND      ;CHECK FOR DONE
014162  002762                          BLT     LOP32D          ;IF NOT, CHECK NEXT SET OF PDR'S
014164  005037  177776                  CLR     @#PS            ;SET MODE TO KERNEL
01[redacted]  005277  164630                  INC     @SR0            ;TURN KT11-D ON
01[redacted]  000005                          RESET                   ;INIT WITH KT11-D ON
```

```
014176  000005                          RESET                   ;INIT WITH KT11-D OFF
014200  012701  001034                  MOV     #ADRTAB,R1      ;R1 REFERENCES ADDRESS OF PDR
014204  012702  000010          LOP32F: MOV     #10,R2          ;R2 KEEPS TRACK OF WHEN TO CHANGE
                                                                ;REGISTER SETS
014210  032771  000100  000000  LOP32G: BIT     #100,@(R1)      ;CHECK W BIT
014216  001001                  BNE     .+4
014220  104006                  HLT                             ;INIT CLEARED W BIT IN PDR WHOSE
                                                                ;ADDRESS IS POINTED TO BY R1
014222  005721                  TST     (R1)+                   ;MOVE POINTER
014224  077207                  SOB     R2,LOP32G               ;CHECK ALL PDR'S IN THIS SET
014226  062701  000020          ADD     #20,R1                  ;CHANGE R1 TO REFERENCE NEXT SET
                                                                ;OF PDR ADDRESSES
014232  020127  001132          CMP     R1,#ADREND              ;CHECK FOR DONE
014236  002762                  BLT     LOP32F                  ;IF NOT, CHECK NEXT SET OF PDR'S
014240  005077  164560          CLR     @SR0                    ;REINITIALIZE SR0

                        ;SHOW THAT A DATO TO A PDR WILL CLEAR THE W BIT
                        ;EVEN WHEN THE INSTRUCTION ALSO CAUSES A TRAP REFERENCE TO
                        ;THE CORRESPONDING PAGE
                        ;MAP KERNEL PAGE 1 RRW AND MAKE A WRITE ACCESS TO PAGE 1
                        ;TO SET THE W BIT
                        ;THEN LOAD THE PDR, MAKING A TRAP REFERENCE TO PAGE 1 IN THE SOURCE
                        ;FETCH OF THE SAME INSTRUCTION-THE W BIT SHOULD BE CLEARED DUE
                        ;TO THE DATO TO THE PDR
014244  104400          TEST56: SCOPE
014246  012706  001000          MOV     #KSTACK,SP              ;INITIALIZE KERNEL STACK POINTER
014252  004767  001562          JSR     PC,ORDER                ;CHECK TEST SEQUENCE + INIT SR0
014256  000056                  56                              ;TEST NUMBER
014260  104006                  HLT                             ;TEST EXECUTED OUT OF SEQUENCE
014262  004767  000552          JSR     %7,RWALL        ;INITIALIZE ALL PAGES RW, BANK 0
014266  004767  001632          JSR     PC,KERN7                ;MAP KERNEL PAR/PDR 7 TO EXT BANK
014272  012777  077406  164576  MOV     #77406,@KPDR1   ;MAKE KERNEL PAGE 1 RRW
014300  012777  000001  164516  MOV     #1,@SR0         ;TURN ON KT11-D
014306  013737  020000  020000  MOV     @#20000,@#20000 ;READ AND WRITE PAGE 1
014314  022777  077506  164554  CMP     #77506,@KPDR1   ;CHECK PDR OF PAGE 1
014322  001401                  BEQ     .+4
014324  104006                  HLT                             ;KERNEL PAGE 1 PDR
                                                                ;INCORRECT - W BIT SHOULD
                                                                ;BE SET DUE TO PREVIOUS MOVE INSTRUCTION
014326  012767  077506  164466  MOV     #77506,TEMP             ;LOAD TEMP WITH VALUE TO BE MOVED TO KPDR1
014334  016777  004462  164534  MOV     TEMP+20000,@KPDR1       ;PAGE 1 REFERENCE SHOULD SET
                                                                ;BUT DATO TO THE PDR CLEARS W BIT
014342  022777  077406  164526  CMP     #77406,@KPDR1   ;CHECK PAGE 1 PDR
014350  001401                  BEQ     .+4
014352  104006                  HLT                             ;PDR INCORRECT - W BIT
014354  005077  164444          CLR     @SR0            ;SHOULD HAVE BEEN CLEARED

                        ;CHECK TO SEE THAT MULTIPLE ACCESSES TO A PAGE AFTER SETTING THE
                        ;W BIT DON'T CLEAR THE W BIT
014360  104400          TEST57: SCOPE
014362  012706  001000          MOV     #KSTACK,SP              ;INITIALIZE KERNEL STACK POINTER
014366  004767  001446          JSR     PC,ORDER                ;CHECK TEST SEQUENCE + INIT SR0
014372  000057                  57                              ;TEST NUMBER
014374  104006                  HLT                             ;TEST EXECUTED OUT OF SEQUENCE
```

```
014376  012767  000010  000706          MOV     #10,ICOUNT
014404  004767  000430                  JSR     X7,RWALL        ;INITIALIZE ALL PAGES 4K, RW, BANK 0
014410  004767  001510                  JSR     PC,KERN7                ;MAP KERNEL PAR/PDR 7 TO EXT BANK
014414  012777  077406  164454          MOV     #77406,@KPDR1   ;MAP KERNEL PAGE 1 RRW
014422  012777  000001  164374          MOV     #1,@SR0         ;TURN ON SEGMENTATION
014430  013737  020000  020000          MOV     @#20000,@#20000 ;READ AND WRITE PAGE 1
014436  022777  077506  164432          CMP     #77506,@KPDR1   ;CHECK THE PDR
014444  001401                          BEQ     .+4
014446  104006                          HLT                     ;KERNEL PDR1 INCORRECT
                                                                ;W BIT SHOULD BE SET

014450  012701  020000                  MOV     #20000,R1
014454  012702  000100                  MOV     #100,R2
014460  005721                  L40:    TST     (R1)+           ;READ PAGE 1 REPEATEDLY
014462  077202                          SOB     R2,L40
014464  022777  077506  164404          CMP     #77506,@KPDR1   ;CHECK W BIT AGAIN
014472  001401                          BEQ     .+4
014474  104006                          HLT                     ;KERNEL PDR 1
                                                                ;INCORRECT AFTER REPEATEDLY READING PAGE 1
014476  005077  164322                  CLR     @SR0            ;TURN OFF SEGMENTATION

                                ;SHOW THAT IF KT11-D IS ON, SETTING THE CURRENT MODE TO 10 WILL
                                ;CAUSE A MEMORY MANAGEMENT ABORT.  NON RESIDENT SHOULD BE SET, AND ALSO PL SHOULD
                                ;BE SET
014502  104400                  TEST60: SCOPE
014504  012706  001000                  MOV     #KSTACK,SP              ;INITIALIZE KERNEL STACK POINTER
014510  004767  001324                  JSR     PC,ORDER                ;CHECK TEST SEQUENCE + INIT SR0
014514  000060                          60                              ;TEST NUMBER
014516  104006                          HLT                             ;TEST EXECUTED OUT OF SEQUENCE
014520  012767  002000  000564          MOV     #2000,ICOUNT    ;RESTORE ITERATION COUNT
014526  004767  000306                  JSR     X7,RWALL
014532  004767  001366                  JSR     PC,KERN7                ;MAP KERNEL PAR/PDR 7 TO EXT BANK
014536  012777  014604  164420          MOV     #RET42,@KTVEC   ;SETUP MEMORY MANAGEMENT ABORT RETURN
014544  005077  164416                  CLR     @KTSTA
014550  012777  000001  164246          MOV     #1,@SR0         ;TURN ON KT11-D
014556  012737  040000  177776  ADD421: MOV     #40000,@#PS     ;SET MODE TO 01-FETCH OF NEXT
014564  000240                          NOP                     ;INSTRUCTION SHOULD ABORT
014566  005077  163204                  CLR     @PS             ;RESTORE MODE TO KERNEL
014572  042777  000001  164224          BIC     #1,@SR0         ;TURN OFF KT11-D
014600  104006                          HLT                     ;NO ABORT WHEN MODE WAS SET
014602  000415                          BR      CONT42          ;TO 01 (ILLEGAL)
014604  042777  000001  164212  RET421: BIC     #1,@SR0         ;TURN OFF KT11-D AFTER ABORT
014612  022777  100040  164204          CMP     #100040,@SR0    ;CK SR0
014620  001401                          BEQ     .+4
014622  104006                          HLT                     ;SR0 INCORRECT AFTER MODE 01 ABORT
                                                                ;NR,AND MODE 01 SHOULD BE SET
014624  022777  014556  164200          CMP     #ADD42,@SR2     ;CHECK SR2
014632  001401                          BEQ     .+4
014634  104006                          HLT                     ;SR2 INCORRECT - SHOULD CONTAIN
                                                                ;ADDRESS OF THE INSTRUCTION
                                                                ;IMMEDIATELY AFTER THE ONE SETTING
                                                                ;THE MODE TO 01
014636  005077  164162          CONT42: CLR     @SR0            ;REINITIALIZE SR0
014642  016777  164320  164314          MOV     KTSTA,@KTVEC    ;RESTORE TRAP CATCHER
```

```
014650  104400                          SCOPE
014652  032767  010000  162710          BIT     #BIT12,SR               ;INHIBIT BELL?
014660  001401                          BEQ     .+4
014662  000406                          BR      LOGICT
014664  012777  000207  164126          MOV     #207,@TDBR
014672  105777  164120                  TSTB    @TCSR
014676  100375                          BPL     .-4
014700  013701  000042          LOGICT: MOV     @#42,R1                         ;MONITOR HOOK
014704  001405                          BEQ     END
014706  000005                          RESET
014710  004711                  LOGIC:  JSR     PC,@R1
014712  000240                          NOP
014714  000240                          NOP
014716  000240                          NOP
014720  000167  164254          END:    JMP     START

                                        ;
                                        ;MESSAGE AREA
014724  005015  052113  030461  MTIT:   .ASCII  <15><12>'KT11-D LOGIC TEST MAINDEC-11-DBKTA-A'<15><12>'@'
014732  042055  046040  043517
014740  041511  052040  051505
014746  020124  040515  047111
014754  042504  026503  030461
014762  042055  045502  040524
014770  040455  005015     100
014775     015  050012  036503  MPC:    .ASCII  <15><12>'PC= @'
015002  040040
015004  020040  051520  020075  MPS:    .ASCII  '  PS= @'
015012     100
        015014                          .EVEN
```

```
                                ;SUBROUTINE TO CLEAR ALL KT11-D REGISTERS (EXCEPT SR1,SR2)
015014  005077  164004          CLRALL: CLR     @SR0
015020  005000                          CLR     R0
015022  012701  000040                  MOV     #32.,R1                 ;COUNT OF REGISTERS TO BE CLEARED
015026  005070  001034          CLRLP:  CLR     @ADRTAB(R0)             ;CLEAR REGISTERS THRU ADDRESS TABLE
015032  005720                          TST     (R0)+                   ;MOVE POINTER
015034  077104                          SOB     R1,CLRLP                ;LOOP TILL DONE
015036  000207                          RTS     X7

                                ;SUBROUTINE TO MAKE ALL PAGES RW, BANK 0, 4K, UP
015040  005077  163760          RWALL:  CLR     @SR0
015044  012701  001034                  MOV     #ADRTAB,R1      ;R1 POINTS TO ADDRESS TABLE
015050  012700  000010          RWL1:   MOV     #10,R0          ;R0 IS COUNTER
015054  005071  000020          RWL2:   CLR     @20(R1)         ;CLEAR PAR
015060  012731  077406                  MOV     #77406,@(R1)+   ;SET PDR RW, 4K
015064  077005                          SOB     R0,RWL2
015066  062701  000020                  ADD     #20,R1
015072  020127  001132                  CMP     R1,#ADREND      ;POINTER TO NEXT GROUP
015076  002764                          BLT     RWL1
015100  000207                          RTS     X7

                                ;ROUTINE TO LOOP THRU A SINGLE INSTRUCTION TEST
                                ;LOAD THE STARTING ADDRESS OF THE TEST
                                ;YOU WISH TO RUN (THE ADDRESS OF THE TESTX
                                ;TAG) AT THE 1ST HALT, SET SWITCH REGISTER
                                ;OPTIONS AT THE 2ND HALT;
                                ;NOTE THAT SW11 MUST BE DOWN AFTER THE 2ND HALT
015102  005037  177776          TESTX:  CLR     @#PS
015106  012706  001000                  MOV     #KSTACK,SP
015112  012737  140000  177776          MOV     #140000,@#PS            ;SETUP USER TRAP
015120  012706  000400                  MOV     #USTACK,SP
015124  005037  177776                  CLR     @#PS
015130  000000                          HALT                            ;WAIT FOR STARTING ADDRESS
015132  016767  162432  000036          MOV     SR,RETRNX               ;LOAD STARTING ADDRESS IN RETRNX
015140  062767  000002  000030          ADD     #2,RETRNX               ;ADD 2 TO POINT TO INSTRUCTION AFTER
015146  000000                          HALT                            ;SET SR OPTIONS
015150  005067  000140                  CLR     SCOPEF                  ;KEEP COUNT AT ZERO
015154  012767  015166  000134          MOV     #XLOOP,RETURN           ;LOAD SCOPE LOOP RETURN POINTER
015162  000177  000010                  JMP     @RETRNX                 ;JUMP TO TEST
015166  005067  000122          XLOOP:  CLR     SCOPEF                  ;KEEP COUNT AT ZERO
015172  000177  000000                  JMP     @RETRNX                 ;JUMP TO TEST
015176  000000          RETRNX: 0
                                ;SCOPE AND/OR ITERATION LOOP FOR EACH TEST 4000 TIMES
015200  032737  040000  177570  SCOPEC: BIT     #BIT14,@#SR             ;TEST SR FOR SCOPE
015206  001015                          BNE     SCOPEB                  ;YES,SCOPE
015210  032737  004000  177570          BIT     #BIT11,@#SR             ;NO-TEST FOR ITERATION
015216  001020                          BNE     SCOPEG                  ;INHIBIT ITERATION
015220  026767  000070  000064          CMP     SCOPEF, ICOUNT          ;COMPARE CURRENT COUNT TO MAX NUMBER
015226  100014                          BPL     SCOPEG                  ;EXIT-DONE
015230  005267  000060                  INC     SCOPEF                  ;INCREMENT COUNT
01[redacted]  012737  000340  177776          MOV     #340,@#[redacted]       ;PREVENT TRAPPING WHILE MOVING STACK
01[redacted]  022626                  SCOPEB: CMP     (6)+,(6[redacted]               ;REPOSITION STACK
```

```
015244  005037  177776                 CLR     @#PS
015250  005077  163550                 CLR     @SR0
015254  000177  000036                 JMP     @RETURN                 ;REPEAT TEST
015260  005067  000030         SCOPEG: CLR     SCOPEF                  ;CLEAR COUNT
015264  005267  163704                 INC     TESTCT                  ;STEP TEST COUNTER TO ALLOW CHECKING
                                                                       ;ORDER OF EXECUTION,
015270  011667  000022                 MOV     @X6,RETURN              ;SAVE SCOPE RETURN POINTER
015274  022626                         CMP     (6)+,(6)+                       ;RETURN INLINE+NEXT TEST
015276  005037  177776                 CLR     @#PS
015302  005077  163516                 CLR     @SR0
015306  000177  000004                 JMP     @RETURN
015312  004000         ICOUNT: 4000                                    ;ITERATION COUNT
015314  000000         SCOPEF: 0                                       ;COUNT LOCATION FOR ITERATION LOOP
015316  000000         RETURN: 0                                       ;ADDRESS OF LAST TEST

                               ;ENTERED WITH SYSTEM TRAP CALL (HLT)
                               ;PRINT OUT THE ERROR PC+2 AND STATUS REGISTER
015320  012767  000340  162450 PRINT: MOV     #340,PS                 ;SET PRIORITY TO 7
015326  032737  020000  177570         BIT     #BIT13,@#SR             ;TEST FOR INHIBIT PRINT OUT
015334  001401                         BEQ     .+4                     ;BRANCH TO PRINT
015336  000430                         BR      CK                      ;INHIBIT, CHECK FOR HALT
015340  012667  000066                 MOV     (6)+,SAVPC              ;PC OF FAILING ROUTINE
015344  012667  000064                 MOV     (6)+,SAVPSR             ;PSR OF ERROR CONDITION
015350  024646                         CMP     -(6),-(6)               ;RESTORE STACK
015352  012767  000200  162416         MOV     #200,PS
015360  016767  000046  000374         MOV     SAVPC,PTEMP1            ;LOAD WITH FAILING PC+2
015366  004767  000044                 JSR     PC,TYPE
015372  014775                         MPC
015374  004767  000116                 JSR     PC,PRSHRT               ;PRINT FAILING PC+2
015400  004767  000032                 JSR     PC,TYPE
015404  015004                         MPS
015406  016767  000022  000346         MOV     SAVPSR,PTEMP1           ;LOAD PROCESSOR STATUS
015414  004767  000130                 JSR     PC,PROCT                ;PRINT PROCESSOR STATUS
015420  005767  162144         CK:     TST     SR                      ;CHECK SR FOR HALT SWITCH
015424  100001                         BPL     .+4                     ;BRANCH IF NOT SET
015426  000000                         HALT                            ;HALT ON ERROR UP
015430  000002                         RTI                             ;RETURN TO MAIN LINE
015432  000000         SAVPC:  0
015434  000000         SAVPSR: 0

                               ;SUBROUTINE TO OUTPUT ASCII MESSAGE ON TELETYPE
015436  010067  000052         TYPE:   MOV     X0,SAVR0
015442  011600                         MOV     (6),X0                  ;GET ADDRESS THAT CONTAINS MESSAGE ADDRESS
015444  062716  000002                 ADD     #2,@X6                  ;SET UP EXIT
015450  011000                         MOV     @X0,X0
015452  112067  000034         TYPA:   MOVB    (0)+,TYPDAT             ;GET CHARACTER
015456  122767  000100  000026         CMPB    #100,TYPDAT             ;CHECK FOR "@" CHARACTER
015464  001003                         BNE     TYPB                    ;BRANCH IF NOT "@"
015466  016700  000022                 MOV     SAVR0,X0                ;RESTORE R0
015472  000207                         RTS     PC                      ;TERMINATOR CHAR, EXIT
015474  116777  000012  163316 TYPB:   MOVB    TYPDAT,@TDBR            ;OUTPUT CHAR TO PRINTER
015502  105777  163310                 TSTB    @TCSR                   ;WAIT FOR TTY READY
015506  100375                         BPL     .-4
015510  000760                         BR      TYPA
```

```
015512  000000                  TYPDAT: 0
015514  000000                  SAVR0:  0
```

```
                                        ;SUBROUTINE TO PRINT OUT OCTAL NUMBER
                                        ;PRSHRT DELETES LEADING ZEROS
                                        ;PROCT PRINTS OUT 6 OCTAL DIGITS
015516  012767  000001  000232  PRSHRT: MOV     #1,PRSFLG       ;SET FLAG TO INDICATE SHORT PRINTOUT
015524  005767  000232          PROCT:  TST     PTEMP1          ;CHECK FOR ZERO
015530  001011                          BNE     PROCT+4         ;BRANCH IF NOT ZERO
015532  012777  000260  163260          MOV     #260,@TDBR      ;OUTPUT A SINGLE ZERO
015540  105777  163252          PROCT1: TSTB    @TCSR           ;WAIT FOR TTY READY
015544  100375                          BPL     .-4
015546  000207                          RTS     %7              ;RETURN
015550  005067  000202          PROCT:  CLR     PRSFLG          ;CLEAR FLAG TO INDICATE FULL PRINTOUT
015554  005067  000206                  CLR     PTEMP3          ;CLEAR R4 FOR COUNTING CHARACTERS OUTPUT
015560  005067  000174                  CLR     PRFLG           ;INITIALIZE CARRY FLAG FOR ROTATES
015564  012767  000260  000172          MOV     #260,PTEMP2     ;SETUP R3
015572  005767  000164          PCK:    TST     PTEMP1          ;CHECK BIT 15 OF NUMBER
015576  100002                          BPL     .+6             ;BRANCH IF ZERO
015600  005267  000160                  INC     PTEMP2          ;INCREMENT R3 IF ONE
015604  006167  000152                  ROL     PTEMP1          ;ROTATE LEFT MOST OCTAL TO RIGHT END
015610  006167  000146                  ROL     PTEMP1
015614  005567  000140                  ADC     PRFLG           ;STORE CARRY
015620  005767  000132          P;CK:   TST     PRSFLG          ;CHECK FOR SHORT PRINTOUT
015624  001404                          BEQ     P;WAIT          ;BRANCH IF NOT SET
015626  026727  000132  000260          CMP     PTEMP2,#260     ;CHECK FOR ZERO IF SET
015634  001410                          BEQ     P;CONT          ;IF SET, GO TO NEXT CHARACTER
015636  016777  000122  163154  P;WAIT: MOV     PTEMP2,@TDBR    ;OUTPUT NEXT CHARACTER
015644  105777  163146          P;WAIT: TSTB    @TCSR           ;WAIT FOR TTY READY
015650  100375                          BPL     .-4
015652  005067  000100                  CLR     PRSFLG          ;PRINT REST OF NUMBER AFTER A NON-ZERO DIGIT
015656  005267  000104          P;CONT: INC     PTEMP3          ;COUNT
015662  026727  000100  000006          CMP     PTEMP3,#6       ;CHECK FOR DONE
015670  001001                          BNE     P;CNT1          ;BRANCH IF NOT DONE
015672  000207                          RTS     %7
015674  000241                  P;CNT1: CLC                     ;CLEAR CARRY
015676  005767  000056                  TST     PRFLG           ;CHECK FOR PREVIOUS CARRY
015702  001403                          BEQ     .+10            ;BRANCH IF PREVIOUSLY ZERO
015704  005067  000050                  CLR     PRFLG           ;INITIALIZE FLAG
015710  000261                          SEC                     ;SET CARRY
015712  006167  000044                  ROL     PTEMP1          ;ROTATE NEXT CHARACTER INTO RIGHT END OF REGISTE
015716  006167  000040                  ROL     PTEMP1
015722  006167  000034                  ROL     PTEMP1
015726  005567  000026                  ADC     PRFLG           ;STORE CARRY
015732  016767  000024  000024          MOV     PTEMP1,PTEMP2   ;LOAD DATA INTO R3
015740  042767  177770  000016          BIC     #177770,PTEMP2  ;CLEAR ALL BUT LOWEST OCTAL DIGIT
015746  052767  000260  000010          BIS     #260,PTEMP2     ;SET TO ASCII EQUIVALENT
015754  000721                          BR      P;CK            ;LOOP
015756  000000          PRSFLG: 0
015760  000000          PRFLG:  0
015762  000000          PTEMP1: 0                               ;CONTAINS VALUE TO BE OUTPUT
015764  000000          PTEMP2: 0                               ;SCRATCH
015766  000000          PTEMP3: 0                               ;USED TO COUNT CHARACTERS OUTPUT
```

```
                                        ;EMT HANDLER
                                        ;FIRST 3 CALLS LEFT OPEN IN TABLE FOR EASY PATCHES
016770  011667  000032          EMTSRV: MOV     @SP,EPC                 ;GET CALL
016774  162767  000002  000024        SUB     #2,EPC
017002  017767  000020  000016        MOV     @EPC,EPC
017010  105067  000013                CLRB    EPC+1                   ;SAVE OFFSET ONLY
017014  062767  016030  000004        ADD     #EMTAB,EPC              ;POINT TO TABLE OF ADDRESSES
017022  017707  000000                MOV     @EPC,PC                 ;JUMP TO DESIRED ROUTINE
017026  000000          EPC:    0
        104000                  PATCH1=EMT+0
        104002                  PATCH2=EMT+2
        104004                  PATCH3=EMT+4
016030  104000          EMTAB:  PATCH1
016032  104002                  PATCH2
016034  104004                  PATCH3
016036  015320                  PRINT
                                        ;
                                        ;SUBROUTINE TO CHECK TEST SEQUENCE
016040  005037  177776          ORDER:  CLR     @#PS                    ;CLEAR PROCESSOR STATUS
016044  011667  000052                  MOV     (SP),TEMPN              ;GET TEST NUMBER ADDRESS
016050  017767  000046  000044        MOV     @TEMPN,TEMPN            ;GET TEST NUMBER
016056  032737  002000  177570        BIT     #BIT10,@#SR
016064  001404                        BEQ     ORDERB
016066  016700  000030                MOV     TEMPN,R0
016072  000005                        RESET
016074  000000                        HALT
016076  026767  163072  000016  ORDERB: CMP     TESTCT,TEMPN            ;IS TEST SEQUENCE CORRECT
016104  001403                        BEQ     ORDERA                  ;YES, CONTINUE
016106  062716  000002                ADD     #2,(SP)                 ;UPDATE FOR ERROR RETURN
016112  000207                        RTS     PC
016114  062716  000004          ORDERA: ADD     #4,(SP)                 ;UPDATE FOR GOOD RETURN
016120  000207                        RTS     PC
016122  000000          TEMPN:  0
                                        ;MAP KERNEL PAR/PDR 7 TO EXTERNAL BANK
016124  012777  007600  163000  KERN7:  MOV     #7600,@KPAR7
016132  012777  077406  162752          MOV     #77406,@KPDR7
016140  000207                        RTS     PC
        017712                          ,=17712

017712  125252          DESTAD: 125252
        000001                  ,END
```

```
ADD23   004650  ADD24   005050  ADD42   014556  ADREND = 001132
ADRTAB  001034  ADR11   012044  ADR2    011004  ADR25   005244
ADR25A  005246  ADR36   007466  ADR36A  007550  ADR36B  007646
ADR36C  007650  ADR37   010022  ADR37A  010104  ADR37B  010202
ADR37C  010204  ADR40   010356  ADR40A  010440  ADR40B  010536
ADR40C  010540  AD13    002514  AD13A   002530  AD13B   002560
AD13C   002610  AD21A = 003742  AD21B = 003754  AD21C   004064
BIT0  = 000001  BIT1  = 000002  BIT10 = 002000  BIT11 = 004000
BIT12 = 010000  BIT13 = 020000  BIT14 = 040000  BIT15 = 100000
BIT2  = 000004  BIT3  = 000010  BIT4  = 000020  BIT5  = 000040
BIT6  = 000100  BIT7  = 000200  BIT8  = 000400  BIT9  = 001000
BLK20   003470  BLOCKS  001176  CK      015420  CKWBIT  013432
CLRALL  015014  CLRLP   015026  CNT22B  004220  CNT22C  004300
CNT22D  004374  CNT22E  004470  CNT31A  005512  CNT31B  005602
CONT11  012110  CONT23  012572  CONT24  013006  CONT42  014636
CONT5   001702  C25     013162  DATA16  003034  DESTAD  017712
DONE13  012254  DONE16  012366  DONE2   011034  DONE23  012604
DONE24  013020  DONE25  013232  DONE36  007724  DONE37  010260
DONE40  010614  DONE5   011466  DONE5A  001720  DONE7   011664
DON25   005362  DST21A  004056  DST21B  004060  DST21C  004062
EMTAB   016030  EMTSRV  015770  END     014720  END20   003554
END22E  004560  END35   007372  EPC     016026  ERR22A  004200
ERR22B  004272  ERR22C  004360  ERR22D  004454  ERR22E  004552
EXIT10  002302  EXIT11  002432  EXIT2   001416  EXIT21  004066
EXIT23  004770  EXIT24  005164  EXT27   013466  FTITLE  001172
HLT   = 104006  ICOUNT  015312  INT25   005256  INT35   004732
INT36   007504  INT36A  007560  INT36B  007660  INT37   010048
INT37A  010114  INT37B  010214  INT40   010374  INT40A  010450
INT40B  010550  IOT35   006530  KERN7   016124  KPAR0   001114
KPAR1   001116  KPAR2   001120  KPAR3   001122  KPAR4   001124
KPAR5   001126  KPAR6   001130  KPAR7   001132  KPDR0   001074
KPDR1   001076  KPDR2   001100  KPDR3   001102  KPDR4   001104
KPDR5   001106  KPDR6   001110  KPDR7   001112  KRET34  006114
KSTACK  001000  KTSTA   001164  KTVEC   001164  K123    001012
K134    001014  LOGIC   014710  LOGICT  014700  LOOP10  002210
LOOP11  002340  LOOP2   001356  LOOP23  012450  LOOP24  012672
LOOP3   001456  LOOP3A  001464  LOOP3B  001470  LOOP52  014064
LOOP4   001546  LOOP4A  001554  LOOP4B  001560  LOOP5   011372
LOOP6   001766  LOOP6A  001774  LOOP7   002072  LOOP7A  002100
LOP10A  002216  LOP10B  002230  LOP10C  002244  LOP11A  002340
LOP11B  002366  LOP11C  002374  LOP27   013364  LOP27A  013376
LOP31A  005460  LOP31B  005464  LOP31C  005516  LOP31D  005530
LOP31E  005560  LOP31F  005564  LOP32C  014102  LOP32D  014130
LOP32E  014134  LOP32F  014204  LOP32G  014210  LOP5A   011450
LOP5AA  001644  LOP5B   001662  L25A    013116  L25B    013126
L40     014460  MPC     014776  MPS     015004  MTIT    014724
NG35B   006506  NG35C   006716  NG35D   007140  NG35E   007356
NOP   = 000240  ODDAD   007164  OK35B   006524  OK35C   006726
OK35D   007154  OK35E   007366  ORDER   016040  ORDERA  016114
ORDERB  016076  PAGES   001156  PAG20   003432  PARTAB  001140
PATCH1 = 104000  PATCH2 = 104002  PATCH3 = 104004  PC    =%000007
PDREND  001136  PDRM2   001170  PDRTAB  001134  PRFLG   015760
PRINT   015320  PROCT   015550  PRSFLG  015756  PRSHRT  015516
PS    = 177776  PTEMP1  015762  PTEMP2  015764  PTEMP3  015766
```

```
P.CK_     015620          P.CNT1    015674          P.CONT    015656          P.WAIT    015636
RETRNX    015176          RETURN    015316          RET11     012056          RET13A    012220
RET16     012356          RET2      011010          RET23A    012516          RET23B    012536
RET24A    012740          RET24B    012752          RET25     013204          RET3      011146
RET33     005742          RET42     014604          RET5      011420          RET7A     011570
RET7C     011644          RWALL     015040          RWL1      015050          RWL2      015056
R0      =%000000          R1      =%000001          R2      =%000002          R3      =%000003
R4      =%000004          R5      =%000005          R6      =%000006          R7      =%000007
SAVEA     001160          SAVEB     001162          SAVPC     015432          SAVPSR    015434
SAVR0     015514          SCOPE   = 104400          SCOPEB    015242          SCOPEC    015200
SCOPEF    015314          SCOPEG    015260          SP      =%000006          SR      = 177570
SR0       001024          SR0H      001026          SR1       001030          SR2       001032
STAEND    001152          STAPNT    001154          START     001200          STATAB    001144
STATUS  = 177776          STAT20    003352          TCSR      001016          TDBR      001020
TEMP      001022          TEMPN     016122          TESTCT    001174          TESTN   = 000061
TESTX     015102          TEST1     001254          TEST10    002144          TEST11    002302
TEST12    002434          TEST13    002476          TEST14    002632          TEST15    002716
TEST16    003036          TEST17    003232          TEST2     001322          TEST20    003622
TEST21    004070          TEST22    004564          TEST23    004772          TEST24    005166
TEST25    005374          TEST26    005662          TEST27    005766          TEST3     001416
TEST30    006252          TEST31    007406          TEST32    007742          TEST33    010276
TEST34    010632          TEST35    010712          TEST36    011054          TEST37    011212
TEST4     001520          TEST40    011314          TEST41    011500          TEST42    011672
TEST43    011764          TEST44    012122          TEST45    012266          TEST46    012400
TEST47    012622          TEST5     001616          TEST50    013036          TEST51    013244
TEST52    013312          TEST53    013466          TEST54    013640          TEST55    014012
TEST56    014244          TEST57    014360          TEST6     001726          TEST60    014502
TEST7     002040          TST10     002200          TST10F    002300          TST11     002330
TST11F    002430          TYPA      015452          TYPB      015474          TYPDAT    015512
TYPE      015436          UPAR0     001054          UPAR1     001056          UPAR2     001060
UPAR3     001062          UPAR4     001064          UPAR5     001066          UPAR6     001070
UPAR7     001072          UPDR0     001034          UPDR1     001036          UPDR2     001040
UPDR3     001042          UPDR4     001044          UPDR5     001046          UPDR6     001050
UPDR7     001052          URET34    006150          USTACK    000400          XLOOP     015166
.       = 017714
```

ERRORS DETECTED:  0

```
*KTLV13,KTLV13=KTLV13,SRC
RUN-TIME: 14 26 0 SECONDS
CORE USED:  5K
```

IDENTIFICATION

| | |
|---|---|
| PRODUCT CODE: | MAINDEC-11-DBKTB-A-D |
| PRODUCT NAME: | KT11-D ACCESS KEYS TEST |
| DATE CREATED: | SEPTEMBER 1, 1972 |
| MAINTAINER: | DIAGNOSTIC GROUP |
| AUTHOR: | ROBERT WHITTON |

COPYRIGHT © 1972
DIGITAL EQUIPMENT CORPORATION

1.0 ABSTRACT

THIS PROGRAM CHECKS THE OPERATION OF EACH ACCESS KEY FOR EACH OF THE FOUR UNIBUS CYCLES (OR COMBINATION OF CYCLES) WHICH MAY REFERENCE AN ADDRESS THRU SEGMENTATION. THESE CYCLES ARE DATI, DATO (NO DATIP), DATIP-DATO, AND DATIP-DATOB. EACH OF THESE CASES IS TESTED WITH AND WITHOUT MEMORY MANAGEMENT ENABLE SET. THUS EIGHT CASES ARE TESTED FOR EACH KEY, SR0, SR1, SR2, THE CORRESPONDING PDR'S, AND THE PROPER EXECUTION OR PREVENTION OF EXECUTION OF THE INSTRUCTION ARE CHECKED IN EACH CASE.

2.0 REQUIREMENTS

2.1 EQUIPMENT

PDP 11/40 WITH KT11-D OPTION

2.2 STORAGE

THE PROGRAM REQUIRES 5K OF MEMORY, STARTING AT LOCATION 0.

3.0 LOADING PROCEDURE

LOAD PROGRAM INTO MEMORY USING ABS LOADER.

4.0 STARTING PROCEDURE

4.1 NORMAL DIAGNOSTIC OPERATION

LOAD ADDRESS 200.
SET DESIRED SWITCH REGISTER SETTINGS (ALL DOWN FOR WORST CASE).
PRESS START.
THE PROGRAM WILL RING THE BELL ON COMPLETION OF A PASS.

4.2 SINGLE SUBTEST LOOP (TESTX)

LOAD ADDRESS 210.
PRESS START.
AT THE FIRST HALT, LOAD THE ADDRESS OF THE DESIRED SUBTEST (THE ADDRESS OF THE TESTXX TAG) INTO THE SWITCH REGISTER, THEN PRESS "CONTINUE".
AT THE SECOND HALT, SET THE OPERATONAL SWITCH SETTINGS DESIRED (SW11 MUST BE SET TO ZERO). THEN PRESS CONTINUE.

## 5.0 OPERATING PROCEDURE

### 5.1 OPERATIONAL SWITCH SETTINGS

SW15=1 OR UP-- HALT ON ERROR
SW14=1 OR UP-- SCOPE LOOP
SW13=1 OR UP-- INHIBIT PRINTOUT
SW11=1 OR UP-- INHIBIT ITERATIONS
SW10=1 OR UP-- HALT AT END OF CURRENT TEST
NEXT TEST NUMBER IN DATA LIGHTS

### 5.2 SUBROUTINE ABSTRACTS

#### 5.2.1 SCOPE

THIS SUBROUTINE CALL IS PLACED BETWEEN EACH SUBTEST. IT RECORDS THE STARTING ADDRESS OF EACH SUB-TEST AS IT IS BEING ENTERED. IF A SCOPE LOOP IS REQUESTED, IT WILL JUMP TO THE START OF THE SUBTEST THAT THE SCOPE LOOP IS REQUESTED FOR. IF SCOPE LOOP IS NOT REQUESTED, THERE WILL BE 1024 ITERATIONS ON THAT SUBTEST BEFORE THE NEXT SUBTEST IS ENTERED. SWITCH 11 ON A 1 INHIBITS ITERATION OF SUBTESTS.

#### 5.2.2 HLT

THIS EMT CALLS THE SUBROUTINE PRINT, WHICH PRINTS OUT THE LOCATION COUNTER AT THE TIME OF FAILURE AND THE CONTENTS OF THE PROCESSOR STATUS REGISTER. NOTE THAT THE LOCATION COUNTER WILL BE THE ADDRESS OF THE HLT PLUS TWO.

#### 5.2.3 TRAPCATCHER

THIS IS A SERIES OF INSTRUCTIONS STARTING AT LOCATION 0 DESIGNED TO DETECT AND ISOLATE UNEXPECTED TRAPS AND INTERRUPTS TO THE TRAP AND INTERRUPT VECTOR AREA OF MEMORY.

IF A HALT OCCURS IN THE TRAP OR INTERRUPT AREA, EXAMINE REGISTER SIX. IT WILL CONTAIN THE CURRENT STACK ADDRESS. THE CONTENTS OF THE CURRENT STACK ADDRESS IS THE VALUE OF THE LOCATION COUNTER WHEN THE TRAP OF INTERRUPT OCCURRED.

#### 5.2.4 TESTX (SINGLE SUBTEST LOOP)

THIS ROUTINE ALLOWS A SINGLE SUBTEST TO BE RUN CONTINUOUSLY FOR SCOPE LOOP PURPOSES. WHILE A SCOPE LOOP SWITCH OPTION EXISTS, IT REQUIRES THAT YOU ARE WITHIN THE TEST IN WHICH YOU WISH TO LOOP. IN SOME CASES (SUCH AS WITH INTERMITTENT FAILURES) THAT'S NOT EASY TO DO. THIS SUBROUTINE ALLOWS YOU TO LOAD THE ADDRESS OF ANY SUBTEST AT THE HALT AND THEN GO DIRECTLY TO THAT TEST.

5.2.5 EMTSRV (EMT DECODER)

THIS ROUTINE DECODES ALL EMT CALLS, INCLUDING PATCHES AND THE HLT CALL WHICH PASSES CONTROL TO THE PRINT ROUTINE.

5.2.6 CLRALL

THIS ROUTINE CLEARS ALL THE PAR'S AND PDR'S OF THE KT11-D, AS WELL AS SR0.

5.2.7 RWALL

THIS ROUTINE MAPS ALL PAGES TO BANK 0 BY CLEARING ALL THE PAR'S, ALL PAGES ARE MADE 4K READ-WRITE BY LOADING ALL THE PDR'S WITH THE VALUE 77406.

5.2.8 SETUP

THIS ROUTINE FIRST CALLS RWALL TO MAP ALL THE PAGES 4K, RW, BANK 0, IT THEN SETS THE KEY FOR KERNEL PAGE 1 TO WHATEVER VALUE WAS STORED ON THE STACK BEFORE THE ROUTINE WAS CALLED. THIS ALLOWS A REFERENCE TO PAGE 1 TO TEST THE DESIRED ACCESS KEY. FINALLY, KERNEL PAGE 7 IS MAPPED TO THE EXTERNAL BANK.

5.3 PROGRAM AND/OR OPERATOR ACTION

5.3.1 SA 200 (NORMAL DIAGNOSTIC OPERATION)

THE PROGRAM EXECUTES SEVERAL TESTS OF EACH KEY.
TESTS 5 THRU 10 ARE CYCLED THRU 3 TIMES,
ONCE FOR EACH OF THE KEYS WHICH GIVES A NON-RESIDENT ABORT,
AT THE END OF EACH PASS THRU THE DIAGNOSTIC THE BELL IS RUNG.

5.3.2 SA 210 (SINGLE SUBTEST LOOP)

THIS STARTING ADDRESS ALLOWS THE USER TO RUN A SINGLE SUBTEST REPEATEDLY BY GIVING THE ADDRESS OF THE DESIRED SUBTEST AT THE FIRST HALT. IF SW11 IS SET TO A ONE, NORMAL TEST EXECUTION WILL BE RESUMED.

6.0 ERRORS

6.1 ERROR PRINTOUT

PRINTOUTS ARE IN A STANDARD TWO-WORD FORMAT. THE FIRST WORD IS THE OCTAL VALUE OF THE PC+2 OF THE DETECTED ERROR. THE SECOND IS THE CONTENTS OF THE PROCESSOR STATUS REGISTER WHEN THE ERROR WAS DETECTED.

6.2 ERROR RECOVERY

IN GENERAL, TEST FAILURES WILL PRINTOUT AN ERROR MESSAGE AND CONTINUE. IF THE "HALT ON ERROR" SWITCH IS SET, HITTING CONTINUE WILL RECOVER. IF THE PROGRAM HANGS UP IN A LOOP, THE ERROR IS LIKELY TO BE A SIGNAL WHICH WAS NEVER RECEIVED. IF A HALT OCCURS IN THE TRAP AND VECTOR AREA THE PROGRAM MUST BE RESTARTED. IF THE PROGRAM HALTS IN THE MAIN FLOW, CONSULT THE LISTING IF NO MESSAGE IS TYPED OUT.

7.0 RESTRICTIONS

PROGRAM MUST BE LOADED INTO LOWER 5K OF MEMORY.

8.0 MISCELLANEOUS

8.1 EXECUTION TIME

EACH PASS TAKES APPROXIMATELY 1 MINUTE WITH CORE MEMORY.

9.0 PROGRAM DESCRIPTION

THE PROGRAM RUNS SEVERAL SEPARATE TESTS OF EACH ACCESS KEY. DATI, DATB (NO DATIP), DATIP-DATO, AND DATIP-DATOB ARE CHECKED FOR EACH KEY, WITH AND WITHOUT MEMORY MANAGEMENT ENABLE SET. THE BELL IS RUNG AT THE END OF EACH PASS.

```
                                ;COPYRIGHT 1972, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
                                ;TEST OF THE KT11-D ACCESS KEYS

                                ;OPERATING INSTRUCTIONS
                                ;       1.  LOAD TEST USING THE ABSOLUTE LOADER
                                ;       2.  LOAD SA 200
                                ;       3.  SET SR TO INITIAL SETTINGS
                                ;       4.  PRESS START

                                ;DYNAMIC SWITCH REGISTER SETTINGS ARE:
                                ;SW15=1 CAUSES HALT ON ERROR
                                ;SW14=1 CAUSES SCOPE LOOPING
                                ;SW13=1 INHIBITS ERROR PRINTOUT
                                ;SW11=1 INHIBITS ITERATIONS
                                ;SW10=1 HALT AT END OF CURRENT TEST WITH NEXT TEST NUMBER
                                ;               IN DATA LIGHTS, PRESS CONTINUE TO ADVANCE TO NEXT TEST.

                                ;DEFINITIONS
104400                                  SCOPE=TRAP
000240                                  NOP=240
000000                                  R0=%0
000001                                  R1=%1
000002                                  R2=%2
000003                                  R3=%3
000004                                  R4=%4
000005                                  R5=%5
000006                                  R6=%6
000007                                  R7=%7
000006                                  SP=%6
000007                                  PC=%7
177570                                  SR=177570
177776                                  PS=177776
177776                                  STATUS=PS
104006                                  HLT=104006

                                ;LOAD TRAP CATCHER IN LOCATIONS 0 THRU 377
                                ;EACH VECTOR ADDRESS IS LOADED WITH THE ADDRESS
                                ;OF THE NEXT LOCATION, AND THE NEXT LOCATION IS LOADED
                                ;WITH A HALT INSTRUCTION (000000)

                                ;LOAD VECTOR AREA
        000030                          .=30
000030  006326                          EMTSRV
000032  000340                          340
        000034                          .=34
000034  005632                          SCOPEC
000036  000000                          0

                                ;LOAD STARTING AREA
        000200                          .=200
000200  000167  001744                  JMP     START
        000210                          .=210
000210  000167  005320                  JMP     TESTX
```

```
                                        ;LOAD DATA AREA
            001P                                .=1000
001000      000(                        KSTACK: 0
            002000                              .=.+776
002000      000000                      USTACK: 0
002002      000000  000000  000000              .WORD 0,0,0,0
002010      000000
002012      177564                      TCSR:   177564          ;TELETYPE PRINTER CSR
002014      177566                      TDBR:   177566
002016      177572                      SR0:    177572          ;KT11-D STATUS REGISTER ADDRESSES
002020      177574                      SR1:    177574
002022      177576                      SR2:    177576
002024      000250                      KTVEC:  250             ;KT11-D INTERRUPT VECTOR
002026      000252                      KTSTA:  252
002030                                  ADRTAB:
002030      177600                      UPDR0:  177600          ;USER PAGE DESCRIPTOR REGISTER ADDRESSES
002032      177602                      UPDR1:  177602
002034      177604                      UPDR2:  177604
002036      177606                      UPDR3:  177606
002040      177610                      UPDR4:  177610
002042      177612                      UPDR5:  177612
002044      177614                      UPDR6:  177614
002046      177616                      UPDR7:  177616
002050      177640                      UPAR0:  177640          ;USER PAGE ADDRESS REGISTER ADDRESSES
002052      177642                      UPAR1:  177642
002054      177644                      UPAR2:  177644
002056      177646                      UPAR3:  177646
002060      177650                      UPAR4:  177650
002062      177652                      UPAR5:  177652
002064      177654                      UPAR6:  177654
002066      177656                      UPAR7:  177656
002070      172300                      KPDR0:  172300          ;KERNEL PAGE DESCRIPTOR REGISTER ADDRESSES
002072      172302                      KPDR1:  172302
002074      172304                      KPDR2:  172304
002076      172306                      KPDR3:  172306
002100      172310                      KPDR4:  172310
002102      172312                      KPDR5:  172312
002104      172314                      KPDR6:  172314
002106      172316                      KPDR7:  172316
002110      172340                      KPAR0:  172340          ;KERNEL PAGE ADDRESS REGISTER ADDRESSES
002112      172342                      KPAR1:  172342
002114      172344                      KPAR2:  172344
002116      172346                      KPAR3:  172346
002120      172350                      KPAR4:  172350
002122      172352                      KPAR5:  172352
002124      172354                      KPAR6:  172354
002126      172356                      KPAR7:  172356
            002126                      ADREND= .-2
002130      000000                      FTITLE: 0               ;TITLE PRINTED FLAG
002132      177573                      SR0H:   177573          ;KT11-D STATUS REGISTER HIGH BYTE ADDRESSES
002134      177575                      SR1H:   177575
002136      177577                      SR2H:   177577
002140      000000                      NRCNT:  0               ;COUNTER FOR TEST OF THE 3 NR KEYS
002142      000000  000004              NRKEYS: 0,4             ;VALUES OF THE 3 NON RESIDENT KEYS
```

```
002146  125252                  DESTAD: 125252                  ;LOCATION USED FOR READS AND WRITES TO CHECK
                                                                ;EXECUTION OR ABORTING AT CORRECT POINT


                                ;SET UP FOR START OF TESTS
002150  005037  177776          START:  CLR     @#PS
002154  012706  001000                  MOV     #KSTACK,SP      ;SETUP KERNEL STACK
002160  012737  140000  177776          MOV     #140000,@#PS    ;SETUP USER STACK POINTER
002166  012706  002000                  MOV     #USTACK,SP
002172  005037  177776                  CLR     @#PS
002176  012767  002000  003520          MOV     #2000,ICOUNT    ;INITIALIZE ITERATION COUNT
002204  012767  002260  003516          MOV     #TEST1+2,RETURN ;SETUP SCOPE AND ITERATION LOOP RETURN
002212  005067  177722                  CLR     NRCNT           ;INITIALIZE FOR NR TEST
002216  012767  000001  004366          MOV     #1,TESTCT       ;SET UP TEST SEQUENCE
002224  005767  177700                  TST     FTITLE          ;TITLE PRINTED
002230  001013                          BNE     TEST1+2         ;YES, SKIP
002232  004767  004156                  JSR     PC,CRLF         ;PRINT TITLE
002236  004767  004204                  JSR     PC,TYPE
002242  005376                          MTIT
002244  004767  004144                  JSR     PC,CRLF
002250  005267  177654                  INC     FTITLE
002254  000401                          BR      .+4
```

```
                                        ;SHOW THAT DATI TO A RRO PAGE (ACF=2) NEITHER TRAPS NOR ABORTS
                                        ;SHOW THAT THE KT11-D STATUS REGISTERS CONTINUE TO TRACK, AND THAT
                                        ;THE PDR CORRESPONDING TO THE REFERENCE IS CORRECT
002256  104400                  TEST1:  SCOPE
002260  012706  011100                  MOV     #KSTACK,SP      ;INITIALIZE KERNEL STACK POINTER
002264  005077  177526                  CLR     @SR0            ;INITIALIZE SR0
002270  004767  004232                  JSR     PC,ORDER        ;CHECK TEST SEQUENCE
002274  000001                          1                       ;TEST NUMBER
002276  104006                          HLT                     ;TEST EXECUTED OUT OF SEQUENCE
002300  012746  000002                  MOV     #2,-(SP)        ;PUSH RRO KEY ON STACK
002304  004767  003174                  JSR     %7,SETUP        ;MAKE KERNEL PAGE 1 RRO, BANK 0
                                                                ;MAKE KERNEL PAGE 7 RW, EXTERNAL
                                                                ;MAKE ALL OTHER PAGES RW, BANK 0
002310  005726                          TST     (SP)+           ;RESTORE STACK
002312  012777  002426  177504          MOV     #RET1,@KTVEC    ;SETUP ABORT RETURN IN CASE
002320  005077  177502                  CLR     @KTSTA
002324  012767  125252  177614          MOV     #125252,DESTAD  ;SETUP LOCATION TO BE REFERENCED
002332  012701  022146                  MOV     #DESTAD+20000,R1        ;R1 CONTAINS VIRTUAL ADDRESS OF LOCATION TO
                                                                ;BE REFERENCED THRU KERNEL PAGE 1
002336  005277  177454                  INC     @SR0            ;TURN ON KT11-D
002342  022721  125252                  CMP     #125252,(R1)+   ;DATI TO RRO PAGE
002346  001404                          BEQ     CMPOK1          ;BRANCH IF CORRECT VALUE WAS READ
002350  005377  177442                  DEC     @SR0            ;ON ERROR, TURN OFF KT11-D
002354  104006                          HLT                     ;RELOCATION FAILED THRU KERNEL PAGE 1
002356  000427                          BR      DONE1
002360  017702  177432          CMPOK1: MOV     @SR0,R2         ;SAVE CONTENTS OF SR0
002364  105377  177426                  DECB    @SR0            ;TURN OFF KT11-D
002370  022702  000017                  CMP     #17,R2          ;CHECK SAVED CONTENTS OF SR0
002374  001401                          BEQ     .+4
002376  104006                          HLT                     ;SR0 INCORRECT-SHOULD HAVE
                                                                ;TRACKED REFERENCE TO PAGE 0,
                                                                ;WHICH GOT THE ADDRESS OF SR0
002400  022777  002400  177414          CMP     #.,@SR2         ;CHECK SR2
002406  001401                          BEQ     .+4
002410  104006                          HLT                     ;SR2 INCORRECT-SHOULD TRACK EVEN
                                                                ;WHEN KT11-D IS OFF
002412  022777  077402  177452          CMP     #77402,@KPDR1   ;CHECK PDR FOR
002420  001401                          BEQ     .+4             ;THE RRO PAGE REFERENCED
002422  104006                          HLT                     ;KPDR1 INCORRECT-SHOULD NOT
                                                                ;HAVE BEEN CHANGED
002424  000404                          BR      DONE1
002426  042777  000001  177362  RET1:   BIC     #1,@SR0         ;TURN OFF KT11-D
002434  104006                          HLT                     ;DATI TO RRO PAGE CAUSED
                                                                ;A TRAP OR ABORT
002436  016777  177364  177360  DONE1:  MOV     KTSTA,@KTVEC    ;RESTORE TRAP RETURN TO CAUSE HALT
002444  005077  177356                  CLR     @KTSTA          ;ON AN UNEXPECTED TRAP
002450  005077  177342                  CLR     @SR0            ;INITIALIZE SR0
002454  005037  177776                  CLR     @#PS            ;INITIALIZE PROCESSOR STATUS

                                        ;SHOW THAT A DATO (NO DATIP) TO A RRO PAGE (ACF=2) ABORTS
                                        ;SHOW THAT THE KT11-D STATUS REGISTERS LOCK UP, AND THAT THE PDR
                                        ;CORRESPONDING TO THE REFERENCE IS CORRECT
002460  104400                  TEST2:  SCOPE
```

```
002462  012706  001000                    MOV     #KSTACK,SP      ;INITIALIZE KERNEL STACK POINTER
002466  005077  177324                    CLR     @SR0            ;INITIALIZE SR0
002472  004767  004030                    JSR     PC,ORDER        ;CHECK TEST SEQUENCE
002476  000002                            2                       ;TEST NUMBER
002500  104006                            HLT                     ;TEST EXECUTED OUT OF SEQUENCE
002502  012746  000002                    MOV     #2,-(SP)        ;PUSH RRO KEY ON STACK
002506  004767  002772                    JSR     %7,SETUP        ;MAKE KERNEL PAGE 1 RRO, BANK 0
                                                                  ;MAKE KERNEL PAGE 7 RW, EXTERNAL
                                                                  ;MAKE ALL OTHER PAGES RW, BANK 0
002512  005726                            TST     (SP)+           ;RESTORE STACK POINTER
002514  012777  002560  177302            MOV     #RET4,@KTVEC    ;SETUP ABORT RETURN
002522  005077  177300                    CLR     @KTSTA
002526  005067  177414                    CLR     DESTAD          ;INITIALIZE LOCATION TO BE ADDRESSED
                                                                  ;BY DATO TO RRO PAGE
002532  012702  022146                    MOV     #DESTAD+20000,R2        ;R2 CONTAINS ADDRESS OF LOCATION
                                                                  ;TO BE REFERENCED THRU KERNEL PAGE 1
002536  012777  000001  177252            MOV     #1,@SR0         ;TURN ON KT11-D
002544  012722  125252          AD4:      MOV     #125252,(R2)+   ;DATO TO RRO PAGE-SHOULD ABORT
002550  005377  177242                    DEC     @SR0            ;TURN OFF KT11-D
002554  104006                            HLT                     ;DATO TO RRO PAGE FAILED TO ABORT
002556  000426                            BR      DONE4
002560  017701  177232          RET4:     MOV     @SR0,R1         ;SAVE CONTENTS OF SR0
002564  005377  177226                    DEC     @SR0            ;TURN OFF KT11-D
002570  022701  020003                    CMP     #20003,R1       ;CHECK SAVED CONTENTS OF SR0
002574  001401                            BEQ     .+4
002576  104006                            HLT                     ;SR0 INCORRECT-SHOULD HAVE LOCKED
                                                                  ;ON DATO TO KERNEL PAGE 1(RR0)
                                                                  ;AND ACCESS FAULT SHOULD BE SET
002600  022777  002544  177214            CMP     #AD4,@SR2       ;CHECK SR2
002606  001401                            BEQ     .+4
002610  104006                            HLT                     ;SR2 INCORRECT-SHOULD HAVE LOCKED
                                                                  ;ON THE ABORTED REFERENCE, WITH THE
                                                                  ;VIRTUAL ADDRESS OF THE INSTRUCTION
002612  022777  077402  177252            CMP     #77402,@KPDR1   ;CHECK INSTRUCTION SPACE PDR
002620  001401                            BEQ     .+4
002622  104006                            HLT                     ;KPDR1 INCORRECT-SHOULD NOT
                                                                  ;HAVE BEEN CHANGED SINCE THE
                                                                  ;DATO DIDN'T WRITE
002624  005767  177316                    TST     DESTAD          ;MAKE CERTAIN THAT DESTINATION
002630  001401                            BEQ     .+4             ;LOCATION WAS NOT WRITTEN
002632  104006                            HLT                     ;DATO TO RRO PAGE WROTE
                                                                  ;INTO THE DESTINATION LOCATION
002634  016777  177166  177162  DONE4:    MOV     KTSTA,@KTVEC    ;CHANGE KT11-D TRAP RETURN
002642  005077  177160                    CLR     @KTSTA          ;TO CAUSE A HALT ON AN UNEXPECTED TRAP
002646  005077  177144                    CLR     @SR0
002652  005037  177776                    CLR     @#PS

                                ;SHOW THAT A DATIP, DATO SEQUENCE TO A RRO PAGE (ACF=2) ABORTS
                                ;SHOW THAT THE KT11-D STATUS REGISTERS LOCK UP, AND THAT THE PDR
                                ;CORRESPONDING TO THE REFERENCE IS CORRECT
002656  104400          TEST3:  SCOPE
002660  012706  001000                    MOV     #KSTACK,SP      ;INITIALIZE KERNEL STACK POINTER
002664  005077  177126                    CLR     @SR0            ;INITIALIZE SR0
002670  004767  003632                    JSR     PC,ORDER        ;CHECK TEST SEQUENCE
```

```
002674  000003                          3                       ;TEST NUMBER
002676  104000                          HLT                     ;TEST EXECUTED OUT OF SEQUENCE
002700  012746  000002                  MOV     #2,-(SP)        ;PUSH RRO KEY ON STACK
002704  004767  002574                  JSR     %7,SETUP        ;MAKE KERNEL PAGE 1 RROT,BANK 0
                                                                ;MAKE KERNEL PAGE 7 RW, EXTERNAL
                                                                ;MAKE ALL OTHER PAGES RW, BANK 0
002710  005726                          TST     (SP)+           ;RESTORE STACK POINTER
002712  012777  002756  177104          MOV     #RET5,@KTVEC    ;SETUP ABORT RETURN
002720  005077  177102                  CLR     @KTSTA
002724  005067  177216                  CLR     DESTAD          ;INITIALIZE LOCATION TO BE ADDRESSED
                                                                ;BY DATIP,DATO TO RRO PAGE
002730  012703  022150                  MOV     #DESTAD+20002,R3        ;R3 CONTAINS VIRTUAL ADDRESS+2 OF LOCATION
                                                                ;TO BE REFERENCED THRU KERNEL PAGE 1
002734  052777  000001  177054          BIS     #1,@SR0         ;TURN ON KT11-D
002742  005243                  AD5:    INC     -(R3)           ;DATIP, DATO TO RRO PAGE
002744  042777  000001  177044          BIC     #1,@SR0         ;TURN OFF KT11-D
002752  104006                          HLT                     ;DATIP, DATO TO RROT PAGE FAILED TO
002754  000427                          BR      DONE5           ;ABORT
002756  017701  177034          RET5:   MOV     @SR0,R1         ;SAVE CONTENTS OF SR0
002762  042777  000001  177026          BIC     #1,@SR0         ;TURN OFF KT11-D
002770  022701  020003                  CMP     #20003,R1       ;CHECK SAVED CONTENTS OF SR0
002774  001401                          BEQ     .+4
002776  104006                          HLT
                                                                ;SR0 INCORRECT-SHOULD HAVE LOCKED
                                                                ;ON DATO TO KERNEL PAGE 1(RR0) AND
                                                                ;ACCESS FAULT SHOULD BE SET
003000  022777  002742  177014          CMP     #AD5,@SR2       ;CHECK SR2
003006  001401                          BEQ     .+4
003010  104006                          HLT
                                                                ;SR2 INCORRECT-SHOULD HAVE LOCKED
                                                                ;ON THE ABORTED REFERENCE, WITH THE
                                                                ;VIRTUAL ADDRESS OF THE INSTRUCTION
003012  022777  077402  177052          CMP     #77402,@KPDR1   ;CHECK PDR
003020  001401                          BEQ     .+4
003022  104006                          HLT
                                                                ;KPDR1 INCORRECT - SHOULD NOT HAVE
                                                                ;BEEN CHANGED, SINCE DATIP IS ABORTED
                                                                ;SINCE IT WILL BE FOLLOWED BY A DATO OR DATOB
003024  005767  177116                  TST     DESTAD          ;MAKE CERTAIN THAT DESTINATION
003030  001401                          BEQ     .+4             ;LOCATION WAS NOT WRITTEN
003032  104006                          HLT                     ;DATO TO RRO PAGE WROTE INTO
                                                                ;THE DESTINATION LOCATION
003034  016777  176766  176762  DONE5:  MOV     KTSTA,@KTVEC    ;CHANGE PAGE FAULT RETURN
003042  005077  176760                  CLR     @KTSTA          ;TO CAUSE A HALT ON AN UNEXPECTED
003046  005077  176744                  CLR     @SR0            ;TRAP
003052  005037  177776                  CLR     @#PS

                                ;SHOW THAT A DATIP,DATOB SEQUENCE TO A RRO PAGE (ACF=2) WORD ABORTS
                                ;SHOW THAT THE KT11-D STATUS REGISTERS LOCK UP, AND THAT THE PDR
                                ;CORRESPONDING TO THE REFERENCE IS CORRECT
003056  104400                  TEST4:  SCOPE
003060  012706  001000                  MOV     #KSTACK,SP      ;INITIALIZE KERNEL STACK POINTER
003064  005077  176726                  CLR     @SR0            ;INITIALIZE SR0
003070  004767  003432                  JSR     PC,ORDER        ;CHECK TEST SEQUENCE
003074  000004                          4                       ;TEST NUMBER
003076  104006                          HLT                     ;TEST EXECUTED OUT OF SEQUENCE
003100  012746  000002                  MOV     #2,-(SP)        ;PUSH RRO KEY ON STACK
```

```
003104  004767  002374                  JSR     %7,SETUP        ;MAKE KERNEL PAGE 1 RRO, BANK 0
                                                                ;MAKE KERNEL PAGE 7 RW, EXTERNAL
                                                                ;MAKE ALL OTHER PAGES RW, BANK 0
003110  005726                          TST     (SP)+           ;RESTORE STACK POINTER
003112  012777  003154  176704          MOV     #RET6,@KTVEC    ;SETUP ABORT RETURN
003120  005077  176702                  CLR     @KTSTA
003124  005067  177016                  CLR     DESTAD          ;INITIALIZE LOCATION TO BE ADDRESSED
                                                                ;BY DATIP,DATOB TO RRO PAGE
003130  012704  022146                  MOV     #DESTAD+20000,R4        ;R4 CONTAINS VIRTUAL ADDRESS OF LOCATION
                                                                ;TO BE REFERENCED THRU KERNEL PAGE 1
003134  052777  000001  176654          BIS     #1,@SR0 ;TURN ON KT11-D
003142  105224                  AD6:    INCB    (R4)+           ;DATIP, DATOB TO RROT PAGE
003144  005377  176646                  DEC     @SR0            ;TURN OFF KT11-D
003150  104006                          HLT                     ;DATIP,DATO TO RROT PAGE FAILED TO ABORT
003152  000426                          BR      DONE6
003154  017701  176636          RET6:   MOV     @SR0,R1         ;SAVE CONTENTS OF SR0
003160  005377  176632                  DEC     @SR0            ;TURN OFF KT11-D
003164  022701  020003                  CMP     #20003,R1       ;CHECK SAVED CONTENTS OF SR0
003170  001401                          BEQ     .+4
003172  104006                          HLT                     ;SR0 INCORRECT-SHOULD HAVE LOCKED ON
                                                                ;DATOB TO KERNEL PAGE 1 (RRO)
                                                                ;ACCESS FAULT SHOULD BE SET
003174  022777  003142  176620          CMP     #AD6,@SR2       ;CHECK SR2
003202  001401                          BEQ     .+4
003204  104006                          HLT                     ;SR2 INCORRECT-SHOULD HAVE LOCKED
                                                                ;ON THE ABORTED REFERENCE, WITH THE
                                                                ;VIRTUAL ADDRESS OF THE INSTRUCTION
003206  022777  077402  176656          CMP     #77402,@KPDR1   ;CHECK PDR
003214  001401                          BEQ     .+4
003216  104006                          HLT                     ;KPDR1 INCORRECT - SHOULD NOT HAVE
                                                                ;BEEN CHANGED-DATIP IS ABORTED
                                                                ;SINCE IT MUST BE FOLLOWED BY A DATO
003220  005767  176722                  TST     DESTAD          ;MAKE CERTAIN THAT DESTINATION
003224  001401                          BEQ     .+4             ;LOCATION WAS NOT WRITTEN
003226  104006                          HLT                     ;DATOB TO RRO PAGE WROTE INTO
                                                                ;THE DESTINATION LOCATION
003230  016777  176572  176566  DONE6:  MOV     KTSTA,@KTVEC    ;CHANGE KT11-D FAULT
003236  005077  176564                  CLR     @KTSTA          ;RETURN TO CAUSE A HALT ON AN
003242  005077  176550                  CLR     @SR0            ;UNEXPECTED TRAP
003246  005037  177776                  CLR     @#PS

                                ;THE FOLLOWING TESTS (5-10) ARE RUN FOR BOTH OF THE NON-RESIDENT
                                ;KEYS - A PASS IS MADE FOR KEY 0, THEN A PASS IS MADE FOR KEY 4,
                                ;THE CURRENT KEY IS STORED ON THE STACK
                                ;SHOW THAT DATI TO A NR PAGE ABORTS WITHOUT COMPLETING
                                ;SHOW THAT THE KT11-D STATUS REGISTERS LOCK UP, AND THAT
                                ;THE PDR CORRESPONDING TO THE REFERENCE IS CORRECT
003252  104400                  TEST5:  SCOPE
003254  012706  001000                  MOV     #KSTACK,SP      ;INITIALIZE KERNEL STACK POINTER
003260  005077  176532                  CLR     @SR0            ;INITIALIZE SR0
003264  004767  003236                  JSR     PC,ORDER        ;CHECK TEST SEQUENCE
003270  000005                          5                       ;TEST NUMBER
003272  104006                          HLT                     ;TEST EXECUTED OUT OF SEQUENCE
003274  005037  001000                  CLR     @#KSTACK        ;PUT 0 ON STACK AS FIRST NR KEY TO BE TESTED
```

```
                                                        ;THIS INSTRUCTION IS SKIPPED WHEN TESTING THE
                                                        ;OTHER WHICH IS SETUP AFTER TEST30
003300  0127..  001000          RERUNA: MOV     #KSTACK,SP
003304  005077    4506                  CLR     @SR0
003310  004767  ( 170                   JSR     %7,SETUP        ;MAKE KERNEL PAGE 1 NR, BANK 0
                                                        ;MAKE KERNEL PAGE 7 RW, EXTERNAL
                                                        ;MAKE ALL OTHER PAGES RW, BANK 0
003314  012777  003360  176502          MOV     #RET21,@KTVEC   ;SETUP ABORT RETURN
003322  005077  176500                  CLR     @KTSTA
003326  005003                          CLR     R3              ;INITIALIZE DESTINATION LOCATION
003330  012767  125252  176610          MOV     #125252,DESTAD  ;INITIALIZE SOURCE LOCATION
003336  012701  022146                  MOV     #DESTAD+20000,R1        ;R1 CONTAINS VIRTUAL ADDRESS OF LOCATION
                                                        ;TO BE REFERENCED THRU KERNEL PAGE 1
003342  005277  176450                  INC     @SR0            ;TURN ON KT11-D
003346  012103                  AD21:   MOV     (R1)+,R3        ;DATI TO NR PAGE - SHOULD ABORT
003350  005377  176442                  DEC     @SR0            ;ON ERROR, TURN OFF KT11-D
003354  104006                          HLT                     ;NO ABORT ON DATI TO A NON-RESIDENT PAGE
003356  000430                          BR      DONE21
003360  017702  176432          RET21:  MOV     @SR0,R2         ;SAVE CONTENTS OF SR0
003364  105377  176426                  DECB    @SR0            ;TURN OFF KT11-D
003370  022702  100003                  CMP     #100003,R2      ;CHECK SAVED CONTENTS OF SR0
003374  001401                          BEQ     .+4
003376  104006                          HLT                     ;SR0 INCORRECT-SHOULD HAVE
                                                        ;LOCKED ON REFERENCE TO
                                                        ;KERNEL PAGE 1 WHICH WAS NON-RESIDENT
003400  022777  003346  176414          CMP     #AD21,@SR2      ;CHECK SR2
003406  001401                          BEQ     .+4
003410  104006                          HLT                     ;SR2 INCORRECT-SHOULD HAVE LOCKED ON
                                                        ;NR REFERENCE
003412  017705  176454                  MOV     @KPDR1,R5       ;MOVE CONTENTS OF KPDR1 TO R5
003416  042705  000007                  BIC     #7,R5           ;TO MASK OFF ACCESS KEY
003422  022705  077400                  CMP     #77400,R5       ;CHECK PDR FOR
003426  001401                          BEQ     .+4             ;THE NR PAGE REFERENCED (BITS 0-2 MASKED OUT)
003430  104006                          HLT                     ;KPDR1 INCORRECT-SHOULD NOT
                                                        ;HAVE BEEN CHANGED
003432  005703                          TST     R3              ;CHECK DESTINATION LOCATION TO SEE
003434  001401                          BEQ     .+4             ;IF INSTRUCTION ALTERRED IT BEFORE ABORTING
003436  104006                          HLT                     ;INSTRUCTION COMPLETED BEFORE ABORT OCCURRED
003440  016777  176362  176356  DONE21: MOV     KTSTA,@KTVEC    ;RESTORE TRAP RETURN TO CAUSE HALT
003446  005077  176354                  CLR     @KTSTA          ;ON AN UNEXPECTED TRAP
003452  005077  176340                  CLR     @SR0            ;INITIALIZE SR0
003456  005037  177776                  CLR     @#PS            ;INITIALIZE PROCESSOR STATUS

                                ;SHOW THAT A DATO (NO DATIP) TO A NR PAGE
                                ;ABORTS WITHOUT COMPLETING THE DATO
                                ;SHOW THAT THE KT11-D STATUS REGISTERS LOCK UP, AND THAT THE PDR
                                ;CORRESPONDING TO THE REFERENCE IS CORRECT
003462  104400                  TEST6:  SCOPE
003464  012706  001000                  MOV     #KSTACK,SP      ;INITIALIZE KERNEL STACK POINTER
003470  005077  176322                  CLR     @SR0            ;INITIALIZE SR0
003474  004767  003026                  JSR     PC,ORDER        ;CHECK TEST SEQUENCE
003500  000006                          6                       ;TEST NUMBER
003502  104006                          HLT                     ;TEST EXECUTED OUT OF SEQUENCE
003504  004767  001774                  JSR     %7,SETUP        ;MAKE KERNEL PAGE 1 NR, BANK 0
```

```
                                                        ;MAKE KERNEL PAGE 7 RW, EXTERNAL
                                                        ;MAKE ALL OTHER PAGES RW, BANK 0
003510  012777  003556  176306          MOV     #RET23,@KTVEC   ;SETUP ABORT RETURN
003516  005077  176304                  CLR     @KTSTA
003522  005067  176420                  CLR     DESTAD          ;INITIALIZE LOCATION TO BE ADDRESSED
                                                        ;BY DATO TO NR PAGE
003526  012701  022146                  MOV     #DESTAD+20000,R1        ;R1 CONTAINS ADDRESS OF LOCATION
                                                        ;TO BE REFERENCED THRU KERNEL PAGE 1
003532  112777  000001  176256          MOVB    #1,@SR0         ;TURN ON KT11-D
003540  012721  125252          AD23:   MOV     #125252,(R1)+   ;DATO TO NR PAGE-SHOULD ABORT
003544  042777  000001  176244          BIC     #1,@SR0         ;TURN OFF KT11-D
003552  104006                          HLT                     ;DATO TO NR PAGE FAILED TO ABORT
003554  000431                          BR      DONE23
003556  017702  176234          RET23:  MOV     @SR0,R2         ;SAVE CONTENTS OF SR0
003562  005377  176230                  DEC     @SR0            ;TURN OFF KT11-D
003566  022702  100003                  CMP     #100003,R2      ;CHECK SAVED CONTENTS OF SR0
003572  001401                          BEQ     .+4
003574  104006                          HLT                     ;SR0 INCORRECT-SHOULD HAVE LOCKED
                                                        ;ON DATO TO KERNEL PAGE 1(NR)
                                                        ;NR FAULT SHOULD BE SET
003576  022777  003540  176216          CMP     #AD23,@SR2      ;CHECK SR2
003604  001401                          BEQ     .+4
003606  104006                          HLT                     ;SR2 INCORRECT-SHOULD HAVE LOCKED
                                                        ;ON THE ABORTED REFERENCE, CONTAINING THE
                                                        ;VIRTUAL ADDRESS OF THE INSTRUCTION
003610  017703  176256                  MOV     @KPDR1,R3       ;MOVE CONTENTS OF KPDR1 TO R3
003614  042703  000007                  BIC     #7,R3           ;TO MASK OFF THE ACCESS KEY
003620  022703  077400                  CMP     #77400,R3       ;CHECK PDR
003624  001401                          BEQ     .+4             ;(BITS 0-2 MASKED OUT)
003626  104006                          HLT                     ;KPDR1 INCORRECT-SHOULD NOT HAVE
                                                        ;BEEN CHANGED
003630  005767  176312                  TST     DESTAD          ;MAKE CERTAIN THAT DESTINATION
003634  001401                          BEQ     .+4             ;LOCATION WAS NOT WRITTEN
003636  104006                          HLT                     ;DATO TO NR PAGE WROTE
                                                        ;INTO THE DESTINATION LOCATION
003640  016777  176162  176156  DONE23: MOV     KTSTA,@KTVEC    ;CHANGE KT11-D FAULT RETURN
003646  005077  176154                  CLR     @KTSTA          ;TO CAUSE A HALT ON AN UNEXPECTED TRAP
003652  005077  176140                  CLR     @SR0
003656  005037  177776                  CLR     @#PS

                                ;SHOW THAT A DATIP, DATO SEQUENCE TO A NR PAGE WORD ABORTS
                                ;SHOW THAT THE KT11-D STATUS REGISTERS LOCK UP, AND THAT THE PDR
                                ;CORRESPONDING TO THE REFERENCE IS CORRECT
003662  104400                  TEST7:  SCOPE
003664  012706  001000                  MOV     #KSTACK,SP      ;INITIALIZE KERNEL STACK POINTER
003670  005077  176122                  CLR     @SR0            ;INITIALIZE SR0
003674  004767  002626                  JSR     PC,ORDER        ;CHECK TEST SEQUENCE
003700  000007                          7                       ;TEST NUMBER
003702  104006                          HLT                     ;TEST EXECUTED OUT OF SEQUENCE
003704  004767  001574                  JSR     %7,SETUP        ;MAKE KERNEL PAGE 1 NR,BANK 0
                                                        ;MAKE KERNEL PAGE 7 RW, EXTERNAL
                                                        ;MAKE ALL OTHER PAGES RW, BANK 0
003710  012777  003754  176106          MOV     #RET25,@KTVEC   ;SETUP ABORT RETURN
003716  005077  176104                  CLR     @KTSTA
```

```
003722  005067  176220                  CLR     DESTAD          ;INITIALIZE LOCATION TO BE ADDRESSED
                                                                ;BY DATIP,DATO TO NR PAGE
003726  012703  022150                  MOV     #DESTAD+20002,R3        ;R3 CONTAINS ADDRESS+2 OF LOCATION
                                                                ;TO BE REFERENCED THRU KERNEL PAGE 1
003732  052777  000001  176056          BIS     #1,@SR0         ;TURN ON KT11-D
003740  005243                  AD25:   INC     -(R3)           ;DATIP, DATO TO NR PAGE-SHOULD ABORT
003742  042777  000001  176046          BIC     #1,@SR0         ;TURN OFF KT11-D
003750  104006                          HLT                     ;DATIP, DATO TO NR PAGE FAILED TO
003752  000432                          BR      DONE25          ;ABORT
003754  017701  176036          RET25:  MOV     @SR0,R1         ;SAVE CONTENTS OF SR0
003760  042777  000001  176030          BIC     #1,@SR0         ;TURN OFF KT11-D
003766  022701  100003                  CMP     #100003,R1      ;CHECK SAVED CONTENTS OF SR0
003772  001401                          BEQ     .+4
003774  104006                          HLT
                                                                ;SR0 INCORRECT-SHOULD HAVE LOCKED
                                                                ;ON DATO TO KERNEL PAGE 1(NR)
                                                                ;NR FAULT SHOULD BE SET
003776  022777  003740  176016          CMP     #AD25,@SR2      ;CHECK SR2
004004  001401                          BEQ     .+4
004006  104006                          HLT
                                                                ;SR2 INCORRECT-SHOULD HAVE LOCKED
                                                                ;ON THE ABORTED REFERENCE, CONTAINING THE
                                                                ;VIRTUAL ADDRESS OF THE INSTRUCTION
004010  017704  176056                  MOV     @KPDR1,R4       ;MOVE CONTENTS OF PDR TO R4
004014  042704  000007                  BIC     #7,R4           ;TO MASK OFF THE ACCESS KEY
004020  022704  077400                  CMP     #77400,R4       ;CHECK PDR
004024  001401                          BEQ     .+4             ;WITH BITS 0-2 MASKED OFF
004026  104006                          HLT                     ;KPDR1 INCORRECT-SHOULD NOT HAVE
                                                                ;BEEN CHANGED
004030  005767  176112                  TST     DESTAD          ;MAKE CERTAIN THAT DESTINATION
004034  001401                          BEQ     .+4             ;LOCATION WAS NOT WRITTEN
004036  104006                          HLT                     ;DATO TO NR PAGE WROTE INTO
                                                                ;THE DESTINATION LOCATION
004040  016777  175762  175756  DONE25: MOV     KTSTA,@KTVEC    ;CHANGE PAGE FAULT RETURN
004046  005077  175754                  CLR     @KTSTA          ;TO CAUSE A HALT ON AN UNEXPECTED
004052  005077  175740                  CLR     @SR0            ;TRAP
004056  005037  177776                  CLR     @#PS
                                ;SHOW THAT A DATIP,DATOB SEQUENCE TO A NR PAGE WORD ABORTS
                                ;SHOW THAT THE KT11-D STATUS REGISTERS LOCK UP, AND THAT THE PDR
                                ;CORRESPONDING TO THE REFERENCE IS CORRECT
004062  104400                  TEST10: SCOPE
004064  012706  001000                  MOV     #KSTACK,SP      ;INITIALIZE KERNEL STACK POINTER
004070  005077  175722                  CLR     @SR0            ;INITIALIZE SR0
004074  004767  002426                  JSR     PC,ORDER        ;CHECK TEST SEQUENCE
004100  000010                          10                      ;TEST NUMBER
004102  104006                          HLT                     ;TEST EXECUTED OUT OF SEQUENCE
004104  004767  001374                  JSR     %7,SETUP        ;MAKE KERNEL PAGE 1 NR, BANK 0
                                                                ;MAKE KERNEL PAGE 7 RW, EXTERNAL
                                                                ;MAKE ALL OTHER PAGES RW, BANK 0
004110  012777  004152  175706          MOV     #RET27,@KTVEC   ;SETUP ABORT RETURN
004116  005077  175704                  CLR     @KTSTA
004122  005067  176020                  CLR     DESTAD          ;INITIALIZE LOCATION TO BE ADDRESSED
                                                                ;BY DATIP,DATOB TO NR PAGE
004126  012704  022146                  MOV     #DESTAD+20000,R4        ;R4 CONTAINS ADDRESS OF LOCATION
                                                                ;TO BE REFERENCED THRU KERNEL PAGE 1
004132  052777  000001  175656          BIS     #1,@SR0         ;TURN ON KT11-D
```

```
004140  105224                  AD27:   INCB    (R4)+           ;DATIP, DATOB TO NR PAGE-SHOULD ABORT
004142  005377  175650                  DEC     @SR0            ;TURN OFF KT11-D
004146  104006                          HLT                     ;DATIP,DATO TO NR PAGE FAILED
004150  000431                          BR      DONE27          ;TO ABORT
004152  017701  175640          RET27:  MOV     @SR0,R1         ;SAVE CONTENTS OF SR0
004156  005377  175634                  DEC     @SR0            ;TURN OFF KT11-D
004162  022701  100003                  CMP     #100003,R1      ;CHECK SAVED CONTENTS OF SR0
004166  001401                          BEQ     .+4
004170  104006                          HLT                     ;SR0 INCORRECT-SHOULD HAVE LOCKED ON
                                                                ;DATIP, DATOB TO KERNEL DATA PAGE 1 (NR)
                                                                ;NR FAULT SHOULD BE SET
004172  022777  004140  175622          CMP     #AD27,@SR2      ;CHECK SR2
004200  001401                          BEQ     .+4
004202  104006                          HLT                     ;SR2 INCORRECT SHOULD HAVE LOCKED
                                                                ;ON THE ABORTED REFERENCE, CONTAINING THE
                                                                ;VIRTUAL ADDRESS OF THE INSTRUCTION
004204  017702  175662                  MOV     @KPDR1,R2       ;MOVE CONTENTS OF PDR 1 TO R2
004210  042702  000007                  BIC     #7,R2           ;TO MASK OFF THE ACCESS KEY
004214  022702  077400                  CMP     #77400,R2       ;CHECK INSTRUCTION SPACE PDR
004220  001401                          BEQ     .+4             ;WITH BITS 0-2 MASKED OFF
004222  104006                          HLT                     ;KPDR1 INCORRECT-SHOULD NOT HAVE
                                                                ;BEEN CHANGED
004224  005767  175716                  TST     DESTAD          ;MAKE CERTAIN THAT DESTINATION
004230  001401                          BEQ     .+4             ;LOCATION WAS NOT WRITTEN
004232  104006                          HLT                     ;DATOB TO NR PAGE WROTE INTO
                                                                ;THE DESTINATION LOCATION
004234  016777  175566  175562  DONE27: MOV     KTSTA,@KTVEC    ;CHANGE KT11-D FAULT
004242  005077  175560                  CLR     @KTSTA          ;RETURN TO CAUSE A HALT ON AN
004246  005077  175544                  CLR     @SR0            ;UNEXPECTED TRAP
004252  005037  177776                  CLR     @#PS
004256  104400                          SCOPE
004260  005267  175654                  INC     NRCNT           ;COUNT HOW MANY NR KEYS HAVE BEEN TESTED
004264  022767  000002  175646          CMP     #2,NRCNT
004272  001416                          BEQ     NXTST           ;IF ALL 2 HAVE BEEN TESTED, BRANCH
004274  016701  175640                  MOV     NRCNT,R1        ;OTHERWISE, CALCULATE OFFSET TO GET NEXT KEY
004300  006301                          ASL     R1
004302  016137  002142  001000          MOV     NRKEYS(R1),@#KSTACK ;PUT NEXT NR KEY ON STACK
004310  012767  003300  001412          MOV     #RERUNA,RETURN          ;PUT NEW SCOPE LOOP ADDRESS IN RETURN
004316  012767  000005  002266          MOV     #5,TESTCT       ;REINIT TEST COUNTER SEQ
004324  000167  176750                  JMP     RERUNA          ;JUMP TO EXECUTE TESTS WITH NEXT NR KEY
004330  005067  175604          NXTST:  CLR     NRCNT
004334  012767  004000  001364          MOV     #4000,SCOPEF
004342  005367  002244                  DEC     TESTCT
004346                          NXTST1:

                                ;SHOW THAT DATI TO A RW PAGE (ACF=6)
                                ;NEITHER TRAPS NOR ABORTS
                                ;SHOW THAT THE KT11-D STATUS REGISTERS CONTINUE TO TRACK, AND THAT
                                ;THE PDR CORRESPONDING TO THE REFERENCE IS CORRECT
004346  104400                  TEST11: SCOPE
004350  012706  001000                  MOV     #KSTACK,SP      ;INITIALIZE KERNEL STACK POINTER
004354  005077  175436                  CLR     @SR0            ;INITIALIZE SR0
004360  004767  002142                  JSR     PC,ORDER        ;CHECK TEST SEQUENCE
004364  000011                          11                      ;TEST NUMBER
```

```
004366  104006                          HLT                     ;TEST EXECUTED OUT OF SEQUENCE
004370  012746  000006                  MOV     #6,-(SP)        ;PUSH RW KEY ON STACK
004374  004767  001104                  JSR     %7,SETUP        ;MAKE KERNEL PAGE 1 RW, BANK 0
                                                                ;MAKE KERNEL PAGE 7 RW, EXTERNAL
                                                                ;MAKE ALL OTHER PAGES RW, BANK 0
004400  005726                          TST     (SP)+           ;RESTORE STACK POINTER
004402  012777  004516  175414          MOV     #RET31,@KTVEC   ;SETUP ABORT RETURN IN CASE
004410  005077  175412                  CLR     @KTSTA
004414  012767  125252  175524          MOV     #125252,DESTAD  ;INITIALIZE LOCATION TO BE READ
004422  012701  022146                  MOV     #DESTAD+20000,R1        ;R1 CONTAINS VIRTUAL ADDRESS OF
                                                                ;LOCATION TO BE REFERENCED THRU KERNEL PAGE 1
004426  005277  175364                  INC     @SR0            ;TURN ON KT11-D
004432  022721  125252                  CMP     #125252,(R1)+   ;DATI TO RW PAGE-SHOULDN'T TRAP OR ABORT
004436  001404                          BEQ     OK31
004440  005377  175352                  DEC     @SR0            ;ON ERROR, TURN OFF KT11-D
004444  104006                          HLT                     ;RELOCATION FAILED THRU KERNEL PAGE 1
004446  000427                          BR      DONE31
004450  017702  175342          OK31:   MOV     @SR0,R2         ;SAVE CONTENTS OF SR0
004454  105377  175336                  DECB    @SR0            ;TURN OFF KT11-D
004460  022702  000017                  CMP     #17,R2          ;CHECK SAVED CONTENTS OF SR0
004464  001401                          BEQ     .+4
004466  104006                          HLT                     ;SR0 INCORRECT-SHOULD HAVE
                                                                ;TRACKED REFERENCE TO
                                                                ;PAGE 0, WHICH GOT THE ADDRESS
                                                                ;OF SR0 TO TURN OFF KT11-D
004470  022777  004470  175324          CMP     #.,@SR2         ;CHECK SR2
004476  001401                          BEQ     .+4
004500  104006                          HLT                     ;SR2 INCORRECT-SHOULD TRACK EVEN
                                                                ;WHEN KT11-D IS OFF
004502  022777  077406  175362          CMP     #77406,@KPDR1   ;CHECK PDR FOR
004510  001401                          BEQ     .+4             ;THE RW PAGE REFERENCED
004512  104006                          HLT                     ;KPDR1 INCORRECT-SHOULD NOT
                                                                ;HAVE BEEN CHANGED
004514  000404                          BR      DONE31
004516  042777  000001  175272  RET31: BIC     #1,@SR0         ;TURN OFF KT11-D
004524  104006                          HLT                     ;DATI TO RW PAGE CAUSED
                                                                ;A TRAP OR ABORT
004526  016777  175274  175270  DONE31: MOV     KTSTA,@KTVEC    ;RESTORE TRAP RETURN TO CAUSE HALT
004534  005077  175266                  CLR     @KTSTA          ;ON AN UNEXPECTED TRAP
004540  005077  175252                  CLR     @SR0            ;INITIALIZE SR0
004544  005037  177776                  CLR     @#PS            ;INITIALIZE PROCESSOR STATUS

                                ;SHOW THAT A DATO (NO DATIP) TO A RW PAGE (ACF=6)
                                ;NEITHER TRAPS NOR ABORTS
                                ;SHOW THAT THE KT11-D STATUS REGISTERS CONTINUE TO TRACK, AND THAT
                                ;THE PDR CORRESPONDING TO THE REFERENCE IS CORRECT
004550  104400                  TEST12: SCOPE
004552  012706  001000                  MOV     #KSTACK,SP      ;INITIALIZE KERNEL STACK POINTER
004556  005077  175234                  CLR     @SR0            ;INITIALIZE SR0
004562  004767  001740                  JSR     PC,ORDER        ;CHECK TEST SEQUENCE
004566  000012                          12                      ;TEST NUMBER
004570  104006                          HLT                     ;TEST EXECUTED OUT OF SEQUENCE
004572  012746  000006                  MOV     #6,-(SP)        ;PUSH RW KEY ON THE STACK
004576  004767  000702                  JSR     %7,SETUP        ;MAKE KERNEL PAGE 1 RW, BANK 0
```

```
                                                        ;MAKE KERNEL PAGE 7 RW, EXTERNAL
                                                        ;MAKE ALL OTHER PAGES RW, BANK 0
004602  005726                          TST     (SP)+   ;RESTORE STACK POINTER
004604  012777  004716  175212          MOV     #RET33,@KTVEC   ;SETUP ABORT RETURN IN CASE
004612  005077  175210                  CLR     @KTSTA
004616  005067  175324                  CLR     DESTAD  ;INITIALIZE LOCATION TO BE REFERENCED
004622  012701  022146                  MOV     #DESTAD+20000,R1        ;R1 CONTAINS VIRTUAL ADDRESS OF
                                                        ;LOCATION TO BE REFERENCED THRU KERNEL PAGE 1
004626  005277  175164                  INC     @SR0    ;TURN ON KT11-D
004632  012721  125252                  MOV     #125252,(R1)+   ;DATO TO RW PAGE-SHOULDN'T TRAP OR ABORT
004636  017702  175154                  MOV     @SR0,R2 ;SAVE CONTENTS OF SR0
004642  105377  175150                  DECB    @SR0    ;TURN OFF KT11-D
004646  022702  000017                  CMP     #17,R2  ;CHECK SAVED CONTENTS OF SR0
004652  001401                          BEQ     .+4
004654  104006                          HLT             ;SR0 INCORRECT-SHOULD HAVE
                                                        ;TRACKED REFERENCE TO DATA SPACE,
                                                        ;PAGE 0, WHICH GOT THE ADDRESS
                                                        ;OF SR0 TO TURN OFF KT11-D
004656  022777  004656  175136          CMP     #.,@SR2 ;CHECK SR2
004664  001401                          BEQ     .+4
004666  104006                          HLT             ;SR2 INCORRECT-SHOULD TRACK EVEN
                                                        ;WHEN KT11-D IS OFF
004670  022777  077506  175174          CMP     #77506,@KPDR1   ;CHECK PDR FOR
004676  001401                          BEQ     .+4     ;THE RW PAGE REFERENCED
004700  104006                          HLT             ;KPDR1 INCORRECT-"W" BIT SHOULD
                                                        ;BE SET SINCE PAGE WAS WRITTEN
004702  022767  125252  175236          CMP     #125252,DESTAD  ;MAKE SURE THAT THE WRITE ACTUALLY OCCURRED
004710  001401                          BEQ     .+4
004712  104006                          HLT             ;DATO TO RW PAGE FAILED TO WRITE CORRECT LOCATION
004714  000404                          BR      DONE33
004716  042777  000001  175072  RET33: BIC     #1,@SR0 ;TURN OFF KT11-D
004724  104006                          HLT             ;DATO TO RW PAGE CAUSED
                                                        ;A TRAP OR ABORT
004726  016777  175074  175070  DONE33: MOV    KTSTA,@KTVEC    ;RESTORE TRAP RETURN TO CAUSE HALT
004734  005077  175066                  CLR     @KTSTA  ;ON AN UNEXPECTED TRAP
004740  005077  175052                  CLR     @SR0    ;INITIALIZE SR0
004744  005037  177776                  CLR     @#PS    ;INITIALIZE PROCESSOR STATUS

                                ;SHOW THAT A DATIP, DATO SEQUENCE TO A RW PAGE (ACF=4)
                                ;NEITHER TRAPS NOR ABORTS
                                ;SHOW THAT THE KT11-D STATUS REGISTERS CONTINUE TO TRACK, AND THAT
                                ;THE PDR CORRESPONDING TO THE REFERENCE IS CORRECT
004750  104400                  TEST13: SCOPE
004752  012706  001000                  MOV     #KSTACK,SP      ;INITIALIZE KERNEL STACK POINTER
004756  005077  175034                  CLR     @SR0    ;INITIALIZE SR0
004762  004767  001540                  JSR     PC,ORDER        ;CHECK TEST SEQUENCE
004766  000013                          13              ;TEST NUMBER
004770  104006                          HLT             ;TEST EXECUTED OUT OF SEQUENCE
004772  012746  000006                  MOV     #6,-(SP)        ;PUSH RW KEY ON THE STACK
004776  004767  000502                  JSR     %7,SETUP        ;MAKE KERNEL PAGE 1 RW, BANK 0
                                                        ;MAKE KERNEL PAGE 7 RW, EXTERNAL
                                                        ;MAKE ALL OTHER PAGES RW, BANK 0
005002  005726                          TST     (SP)+   ;RESTORE STACK POINTER
005004  012777  005114  175012          MOV     #RET35,@KTVEC   ;SETUP ABORT RETURN IN CASE
```

```
005012  005077  175010                  CLR     @KTSTA
005016  005067  175124                  CLR     DESTAD          ;INITIALIZE LOCATION TO BE REFERENCED
005022  012704  022150                  MOV     #DESTAD+20002,R4        ;R4 CONTAINS VIRTUAL ADDRESS+2 OF
                                                                ;LOCATION TO BE REFERENCED THRU KERNEL PAGE 1
005026  005277  174764                  INC     @SR0            ;TURN ON KT11-D
005032  005244                          INC     -(R4)           ;DATIP, DATO TO RW PAGE-SHOULDN'T TRAP OR ABORT
005034  017702  174756                  MOV     @SR0,R2         ;SAVE CONTENTS OF SR0
005040  105077  174752                  CLRB    @SR0            ;TURN OFF KT11-D
005044  022702  000017                  CMP     #17,R2          ;CHECK SAVED CONTENTS OF SR0
005050  001401                          BEQ     .+4
005052  104006                          HLT                     ;SR0 INCORRECT-SHOULD HAVE
                                                                ;TRACKED REFERENCE TO DATA SPACE,
                                                                ;PAGE 0, WHICH GOT THE ADDRESS
                                                                ;OF SR0 TO TURN OFF KT11-D
005054  022777  005054  174740          CMP     #.,@SR2         ;CHECK SR2
005062  001401                          BEQ     .+4
005064  104006                          HLT                     ;SR2 INCORRECT-SHOULD TRACK EVEN
                                                                ;WHEN KT11-D IS OFF
005066  022777  077506  174776          CMP     #77506,@KPDR1   ;CHECK PDR CORRESPONDING
005074  001401                          BEQ     .+4             ;TO THE RW REFERENCE
005076  104006                          HLT                     ;KPDR1 INCORRECT - "W" BIT SHOULD BE SET
005100  022767  000001  175040          CMP     #1,DESTAD       ;MAKE CERTAIN THAT THE INSTRUCTION WAS EXECUTED
005106  001401                          BEQ     .+4
005110  104006                          HLT                     ;DATIP, DATO TO RW PAGE DIDN'T EXECUTE CORRECTLY
005112  000404                          BR      DONE35
005114  042777  000001  174674  RET35:  BIC     #1,@SR0         ;TURN OFF KT11-D
005122  104006                          HLT                     ;DATIP, DATO TO RW PAGE CAUSED
                                                                ;A TRAP OR ABORT
005124  016777  174676  174672  DONE35: MOV     KTSTA,@KTVEC    ;RESTORE TRAP RETURN TO CAUSE HALT
005132  005077  174670                  CLR     @KTSTA          ;ON AN UNEXPECTED TRAP
005136  005077  174654                  CLR     @SR0            ;INITIALIZE SR0
005142  005037  177776                  CLR     @#PS            ;INITIALIZE PROCESSOR STATUS


                                ;SHOW THAT A DATIP, DATOB SEQUENCE TO A RW PAGE (ACF=6)
                                ;NEITHER TRAPS NOR ABORTS
                                ;SHOW THAT THE KT11-D STATUS REGISTERS CONTINUE TO TRACK, AND THAT
                                ;THE PDR CORRESPONDING TO THE REFERENCE IS CORRECT
005146  104400                  TEST14: SCOPE
005150  012706  001000                  MOV     #KSTACK,SP      ;INITIALIZE KERNEL STACK POINTER
005154  005077  174636                  CLR     @SR0            ;INITIALIZE SR0
005160  004767  001342                  JSR     PC,ORDER        ;CHECK TEST SEQUENCE
005164  000014                          14                      ;TEST NUMBER
005166  104006                          HLT                     ;TEST EXECUTED OUT OF SEQUENCE
005170  012746  000006                  MOV     #6,-(SP)        ;PUSH RW KEY ON THE STACK
005174  004767  000304                  JSR     %7,SETUP        ;MAKE KERNEL PAGE 1 RW, BANK 0
                                                                ;MAKE KERNEL PAGE 7 RW, EXTERNAL
                                                                ;MAKE ALL OTHER PAGES RW, BANK 0
005200  005726                          TST     (SP)+           ;RESTORE STACK POINTER
005202  012777  005312  174614          MOV     #RET37,@KTVEC   ;SETUP ABORT RETURN IN CASE
005210  005077  174612                  CLR     @KTSTA
005214  005067  174726                  CLR     DESTAD          ;INITIALIZE LOCATION TO BE REFERENCED
005220  012703  022147                  MOV     #DESTAD+20001,R3        ;R3 CONTAINS VIRTUAL ADDRESS+1 OF
                                                                ;LOCATION TO BE REFERENCED THRU KERNEL PAGE 1
```

```
005224  005277  174566                  INC     @SR0            ;TURN ON KT11-D
005230  105343                          DECB    -(R3)           ;DATIP, DATOB TO RW PAGE-SHOULDN'T TRAP OR ABORT
005232  017702  174560                  MOV     @SR0,R2         ;SAVE CONTENTS OF SR0
005236  105377  174554                  DECB    @SR0            ;TURN OFF KT11-D
005242  022702  000017                  CMP     #17,R2          ;CHECK SAVED CONTENTS OF SR0
005246  001401                          BEQ     .+4
005250  104006                          HLT                     ;SR0 INCORRECT-SHOULD HAVE
                                                                ;TRACKED REFERENCE TO DATA SPACE,
                                                                ;PAGE 0, WHICH GOT THE ADDRESS
                                                                ;OF SR0 TO TURN OFF KT11-D
005252  022777  005252  174542          CMP     #.,@SR2         ;CHECK SR2
005260  001401                          BEQ     .+4
005262  104006                          HLT                     ;SR2 INCORRECT-SHOULD TRACK EVEN
                                                                ;WHEN KT11-D IS OFF
005264  022777  077506  174600          CMP     #77506,@KPDR1   ;CHECK PDR CORRESPONDING
005272  001401                          BEQ     .+4             ;TO THE RW REFERENCE
005274  104006                          HLT                     ;KPDR1 INCORRECT - "W" BIT SHOULD BE SET
005276  022767  000377  174642          CMP     #377,DESTAD     ;MAKE CERTAIN THAT THE INSTRUCTION WAS EXECUTED
005304  001401                          BEQ     .+4
005306  104006                          HLT
005310  000404                          BR      DONE37          ;DATIP, DATOB TO RW PAGE DIDN'T EXECUTE CORRECTLY
005312  042777  000001  174476  RET37:  BIC     #1,@SR0         ;TURN OFF KT11-D
005320  104006                          HLT                     ;DATIP, DATOB TO RW PAGE CAUSED
                                                                ;A TRAP OR ABORT
005322  016777  174500  174474  DONE37: MOV     KTSTA,@KTVEC    ;RESTORE TRAP RETURN TO CAUSE HALT
005330  005077  174472                  CLR     @KTSTA          ;ON AN UNEXPECTED TRAP
005334  005077  174456                  CLR     @SR0            ;INITIALIZE SR0
005340  005037  177776                  CLR     @#PS            ;INITIALIZE PROCESSOR STATUS

005344  104400                          SCOPE

005346  004767  001024                  JSR     %7,BELL

005352  013701  000042                  MOV     @#42,R1         ;MONITOR HOOK
005356  001405                          BEQ     END
005360  000005                          RESET
005362  004711                  LOGIC:  JSR     %7,@R1
005364  000240                          NOP
005366  000240                          NOP
005370  000240                          NOP
005372  000167  174552          END:    JMP     START
```

```
                                        ;
                                        ;MESSAGE AREA
005376  052113  030461  042055   MTIT:   .ASCII  'KT11-D ACCESS KEYS TEST@'
005404  040440  041503  051505
005412  020123  042513  051531
005420  052040  051505  040124
005426  041520  020075     100   MPC:    .ASCII  'PC= @'
005433     040  050040  036523   MPS:    .ASCII  '  PS= @'
005440  040040
                                        .EVEN
                                        ;SUBROUTINE TO MAKE ALL PAGES RW, BANK 0, 4K, UP
005442  005077  174350            RWALL:  CLR     @SR0
005446  012701  002030                    MOV     #ADRTAB,R1
005452  012700  000010            RWL1:   MOV     #10,R0
005456  005071  000020            RWL2:   CLR     @20(R1)
005462  012731  077406                    MOV     #77406,@(R1)+
005466  077005                            SOB     R0,RWL2
005470  062701  000020                    ADD     #20,R1
005474  020127  002126                    CMP     R1,#ADREND
005500  003764                            BLE     RWL1
005502  000207                            RTS     %7

                                        ;SUBROUTINE TO SET ALL PAGES RW EXCEPT KERNEL PAGE 1
                                        ;KERNEL PAGE 1 IS SET TO DESIRED KEY
                                        ;KEY IS PASSED VIA THE STACK
                                        ;ALL PAGES ARE MAPPED TO BANK 0 EXCEPT KERNEL PAGE 7, WHICH IS MAPPED TO
                                        ;THE EXTERNAL BANK
005504  004767  177732            SETUP:  JSR     %7,RWALL        ;INITIALLY MAP ALL PAGES RW, BANK 0
005510  012777  077400  174354            MOV     #77400,@KPDR1   ;MAKE KERNEL PAGE ONE 4K, UP
005516  056677  000002  174346            BIS     2(SP),@KPDR1    ;SET TO DESIRED KEY
005524  012777  007600  174374            MOV     #7600,@KPAR7    ;MAP KERNEL PAGE 7 EXTERNAL
005532  000207                            RTS     %7
                                        ;ROUTINE TO LOOP THRU A SINGLE INSTRUCTION TEST
                                        ;LOAD THE STARTING ADDRESS OF THE TEST
                                        ;YOU WISH TO RUN (THE ADDRESS OF THE TESTX
                                        ;TAG) AT THE 1ST HALT, SET SWITCH REGISTER
                                        ;OPTIONS AT THE 2ND HALT.
                                        ;NOTE THAT SW11 MUST BE DOWN AFTER THE 2ND HALT
005534  005037  177776            TESTX:  CLR     @#PS
005540  012706  001000                    MOV     #KSTACK,SP
005544  012737  140000  177776            MOV     #140000,@#PS    ;SETUP USER STACK POINTER
005552  012706  002000                    MOV     #USTACK,SP
005556  005037  177776                    CLR     @#PS
005562  000000                            HALT                    ;WAIT FOR STARTING ADDRESS
005564  016767  172000  000036            MOV     SR,RETRNX       ;LOAD STARTING ADDRESS IN RETRNX
005572  062767  000002  000030            ADD     #2,RETRNX       ;ADD 2 TO POINT TO INSTRUCTION AFTER
005600  000000                            HALT                    ;SET SR OPTIONS
005602  005067  000120                    CLR     SCOPEF          ;KEEP COUNT AT ZERO
005606  012767  005620  000114            MOV     #XLOOP,RETURN   ;LOAD SCOPE LOOP RETURN POINTER
005614  000177  000010                    JMP     @RETRNX         ;JUMP TO TEST
005620  005067  000102            XLOOP:  CLR     SCOPEF          ;KEEP COUNT AT ZERO
005624  000177  000000                    JMP     @RETRNX         ;JUMP TO TEST
005630  000000                    RETRNX: 0
```

```
                                        ;SCOPE AND/OR ITERATION LOOP FOR EACH TEST 4000 TIMES
005632  032737  040000  177570  SCOPEC: BIT     #40000,@#SR     ;TEST SR FOR SCOPE
005640  001015                          BNE     SCOPEB          ;YES,SCOPE
005642  032737  004000  177570          BIT     #4000,@#SR      ;NO-TEST FOR ITERATION
005650  001016                          BNE     SCOPEG          ;INHIBIT ITERATION
005652  026767  000050  000044          CMP     SCOPEF, ICOUNT  ;COMPARE CURRENT COUNT TO MAX NUMBER
005660  100012                          BPL     SCOPEG          ;EXIT-DONE
005662  005267  000040                  INC     SCOPEF          ;INCREMENT COUNT
005666  012737  000340  177776          MOV     #340,@#PS       ;PREVENT TRAPPING WHILE MOVING STACK
005674  022606                  SCOPEB: CMP     (6)+,%6         ;REPOSITION STACK
005676  012637  177776                  MOV     (6)+,@#PS       ;RESTORE PREVIOUS PROCESSOR STATUS
005702  000177  000022                  JMP     @RETURN         ;REPEAT TEST
005706  005067  000014          SCOPEG: CLR     SCOPEF          ;CLEAR COUNT
005712  005267  000674                  INC     TESTCT          ;STEP TEST COUNTER
005716  011667  000006                  MOV     @%6,RETURN      ;SAVE SCOPE RETURN POINTER
005722  000002                          RTI                     ;RETURN INLINE-NEXT TEST
005724  004000                  ICOUNT: 4000                    ;ITERATION COUNT
005726  000000                  SCOPEF: 0                       ;COUNT LOCATION FOR ITERATION LOOP
005730  000000                  RETURN: 0                       ;ADDRESS OF LAST TEST

                                        ;ENTERED WITH SYSTEM TRAP CALL (HLT)
                                        ;PRINT OUT THE ERROR PC+2 AND STATUS REGISTER
005732  012767  000340  172036  PRINT:  MOV     #340,PS         ;SET PRIORITY TO 7
005740  036727  171624  020000          BIT     SR,#20000       ;TEST FOR INHIBIT PRINT OUT
005746  001401                          BEQ     .+4             ;BRANCH TO PRINT
005750  000432                          BR      CK              ;INHIBIT, CHECK FOR HALT
005752  012667  000072                  MOV     (6)+,SAVPC      ;PC OF FAILING ROUTINE
005756  012667  000070                  MOV     (6)+,SAVPSR     ;PSR OF ERROR CONDITION
005762  024646                          CMP     -(6),-(6)       ;RESTORE STACK
005764  012767  000200  172004          MOV     #200,PS
005772  004767  000416                  JSR     %7,CRLF         ;OUTPUT CARRIAGE RETURN AND LINE FEED
005776  016767  000046  000314          MOV     SAVPC,PTEMP1    ;LOAD WITH FAILING PC+2
006004  004767  000436                  JSR     PC,TYPE
006010  005426                          MPC
006012  004767  000036                  JSR     PC,PRSHRT
006016  004767  000424                  JSR     PC,TYPE
006022  005433                          MPS
006024  016767  000022  000266          MOV     SAVPSR,PTEMP1   ;LOAD PROCESSOR STATUS
006032  004767  000050                  JSR     %7,PROCT        ;PRINT PROCESSOR STATUS
006036  005767  171526          CK:     TST     SR              ;CHECK SR FOR HALT SWITCH
006042  100001                          BPL     .+4             ;BRANCH IF NOT SET
006044  000000                          HALT                    ;HALT ON ERROR UP
006046  000002                          RTI                     ;RETURN TO MAIN LINE
006050  000000                  SAVPC:  0
006052  000000                  SAVPSR: 0
```

```
                                        ;SUBROUTINE TO PRINT OUT OCTAL NUMBER
                                        ;PRSHRT DELETES LEADING ZEROS
                                        ;PROCT PRINTS OUT 6 OCTAL DIGITS
006054  012767  000001  000232  PRSHRT: MOV     #1,PRSFLG       ;SET FLAG TO INDICATE SHORT PRINTOUT
006062  005767  000232                  TST     PTEMP1          ;CHECK FOR ZERO
006066  001011                          BNE     PROCT+4         ;BRANCH IF NOT ZERO
006070  012777  000260  173716          MOV     #260,@TDBR      ;OUTPUT A SINGLE ZERO
006076  105777  173710                  TSTB    @TCSR           ;WAIT FOR TTY READY
006102  100375                          BPL     .-4
006104  000207                          RTS     %7              ;RETURN
006106  005067  000202          PROCT:  CLR     PRSFLG          ;CLEAR FLAG TO INDICATE FULL PRINTOUT
006112  005067  000206                  CLR     PTEMP3          ;CLEAR R4 FOR COUNTING CHARACTERS OUTPUT
006116  005067  000174                  CLR     PRFLG           ;INITIALIZE CARRY FLAG FOR ROTATES
006122  012767  000260  000172          MOV     #260,PTEMP2     ;SETUP R3
006130  005767  000164                  TST     PTEMP1          ;CHECK BIT 15 OF NUMBER
006134  100002                          BPL     .+6             ;BRANCH IF ZERO
006136  005267  000160                  INC     PTEMP2          ;INCREMENT R3 IF ONE
006142  006167  000152                  ROL     PTEMP1          ;ROTATE LEFT MOST OCTAL TO RIGHT END
006146  006167  000146                  ROL     PTEMP1
006152  005567  000140                  ADC     PRFLG           ;STORE CARRY
006156  005767  000132          P.CK:   TST     PRSFLG          ;CHECK FOR SHORT PRINTOUT
006162  001404                          BEQ     P.WAIT          ;BRANCH IF NOT SET
006164  026727  000132  000260          CMP     PTEMP2,#260     ;CHECK FOR ZERO IF SET
006172  001410                          BEQ     P.CONT          ;IF SET, GO TO NEXT CHARACTER
006174  016777  000122  173612  P.WAIT: MOV     PTEMP2,@TDBR    ;OUTPUT NEXT CHARACTER
006202  105777  173604                  TSTB    @TCSR           ;WAIT FOR TTY READY
006206  100375                          BPL     .-4
006210  005067  000100                  CLR     PRSFLG          ;PRINT REST OF NUMBER AFTER A NON-ZERO DIGIT
006214  005267  000104          P.CONT: INC     PTEMP3          ;COUNT
006220  026727  000100  000006          CMP     PTEMP3,#6       ;CHECK FOR DONE
006226  001001                          BNE     P.CNT1          ;BRANCH IF NOT DONE
006230  000207                          RTS     %7
006232  000241          P.CNT1: CLC                             ;CLEAR CARRY
006234  005767  000056                  TST     PRFLG           ;CHECK FOR PREVIOUS CARRY
006240  001403                          BEQ     .+10            ;BRANCH IF PREVIOUSLY ZERO
006242  005067  000050                  CLR     PRFLG           ;INITIALIZE FLAG
006246  000261                          SEC                     ;SET CARRY
006250  006167  000044                  ROL     PTEMP1          ;ROTATE NEXT CHARACTER INTO RIGHT END OF REGISTER
006254  006167  000040                  ROL     PTEMP1
006260  006167  000034                  ROL     PTEMP1
006264  005567  000026                  ADC     PRFLG           ;STORE CARRY
006270  016767  000024  000024          MOV     PTEMP1,PTEMP2   ;LOAD DATA INTO R3
006276  042767  177770  000016          BIC     #177770,PTEMP2  ;CLEAR ALL BUT LOWEST OCTAL DIGIT
006304  052767  000260  000010          BIS     #260,PTEMP2     ;SET TO ASCII EQUIVALENT
006312  000721                          BR      P.CK            ;LOOP
006314  000000          PRSFLG: 0
006316  000000          PRFLG:  0
006320  000000          PTEMP1: 0                               ;CONTAINS VALUE TO BE OUTPUT
006322  000000          PTEMP2: 0                               ;SCRATCH
006324  000000          PTEMP3: 0                               ;USED TO COUNT CHARACTERS OUTPUT
```

```
                                        ;EMT HANDLER
                                        ;FIRST 3 CALLS LEFT OPEN IN TABLE FOR EASY PATCHES
006326  011667  000032          EMTSRV: MOV     @SP,EPC         ;GET CALL
006332  162767  000002  000024        SUB     #2,EPC
006340  017767  000020  000016        MOV     @EPC,EPC
006346  105067  000013                CLRB    EPC+1           ;SAVE OFFSET ONLY
006352  062767  006366  000004        ADD     #EMTAB,EPC      ;POINT TO TABLE OF ADDRESSES
006360  017707  000000                MOV     @EPC,PC         ;JUMP TO DESIRED ROUTINE
006364  000000          EPC:    0
        000000                  PATCH1=0                        ;SUBSTITUTE 104000 WHERE 1ST PATCH IS NEEDED
        000000                  PATCH2=0                        ;104002 FOR 2ND PATCH
        000000                  PATCH3=0                        ;104004 FOR 3RD PATCH
006366  000000          EMTAB:  PATCH1                          ;LOAD ADDRESS OF 1ST PATCH HERE
006370  000000                  PATCH2                          ;LOAD ADDRESS OF 2ND PATCH HERE
006372  000000                  PATCH3                          ;LOAD ADDRESS OF 3RD PATCH HERE
006374  005732                  PRINT


                                        ;BELL ON PASS COMPLETE
006376  012777  000207  173410  BELL:   MOV     #207,@TDBR
006404  105777  173402                  TSTB    @TCSR
006410  100375                        BPL     .-4
006412  000207                        RTS     %7

                                        ;SUBROUTINE TO OUTPUT CARRIAGE RETURN AND LINEFEED
006414  012777  000215  173372  CRLF:   MOV     #215,@TDBR      ;OUTPUT CARRIAGE RETURN
006422  105777  173364                  TSTB    @TCSR           ;WAIT FOR TTY READY
006426  100375                        BPL     .-4
006430  012777  000212  173356        MOV     #212,@TDBR      ;OUTPUT LINEFEED
006436  105777  173350                  TSTB    @TCSR           ;WAIT FOR TTY READY
006442  100375                        BPL     .-4
006444  000207                        RTS     %7              ;RETURN
```

```
                                        ;SUBROUTINE TO OUTPUT ASCII MESSAGE ON TELETYPE
006446  010067  000052          TYPE:   MOV     %0,SAVR0
006452  011600                          MOV     (6),%0          ;GET ADDRESS THAT CONTAINS MESSAGE ADDRESS
006454  062716  000002                  ADD     #2,@%6          ;SET UP EXIT
006460  011000                          MOV     @%0,%0
006462  112067  000034          TYPA:   MOVB    (0)+,TYPDAT     ;GET CHARACTER
006466  122767  000100  000026          CMPB    #100,TYPDAT     ;CHECK FOR "@" CHARACTER
006474  001003                          BNE     TYPB            ;BRANCH IF NOT "@"
006476  016700  000022                  MOV     SAVR0,%0        ;RESTORE R0
006502  000207                          RTS     PC              ;TERMINATOR CHAR, EXIT
006504  116777  000012  173302  TYPB:   MOVB    TYPDAT,@TDBR    ;OUTPUT CHAR TO PRINTER
006512  105777  173274                  TSTB    @TCSR           ;WAIT FOR TTY READY
006516  100375                          BPL     .-4
006520  000760                          BR      TYPA
006522  000000                  TYPDAT: 0
006524  000000                  SAVR0:  0
                                        ;
                                        ;SUBROUTINE TO CHECK TEST SEQUENCE
006526  005037  177776          ORDER:  CLR     @#PS            ;CLEAR PROCESSOR STATUS
006532  011667  000052                  MOV     (SP),TEMPN      ;GET TEST NUMBER ADDRESS
006536  017767  000046  000044          MOV     @TEMPN,TEMPN    ;GET TEST NUMBER
006544  032737  002000  177570          BIT     #2000,@#SR
006552  001404                          BEQ     ORDERB
006554  016700  000030                  MOV     TEMPN,R0
006560  000005                          RESET
006562  000000                          HALT
006564  026767  000022  000016  ORDERB: CMP     TESTCT,TEMPN    ;IS TEST SEQUENCE CORRECT
006572  001403                          BEQ     ORDERA          ;YES, CONTINUE
006574  062716  000002                  ADD     #2,(SP)         ;UPDATE FOR ERROR RETURN
006600  000207                          RTS     PC
006602  062716  000004          ORDERA: ADD     #4,(SP)         ;UPDATE FOR GOOD RETURN
006606  000207                          RTS     PC
006610  000000                  TEMPN:  0
006612  000000                  TESTCT: 0
        000001                          .END
```

```
ADREND = 002126        ADRTAB  002030        AD21    003346        AD23    003540
AD25    003740        AD27    004140        AD4     002544        AD5     002742
AD6     003142        BELL    006376        CK      006036        CMPOK1  002360
CRLF    006414        DESTAD  002146        DONE1   002436        DONE21  003440
DONE23  003640        DONE25  004040        DONE27  004234        DONE3   004526
DONE33  004726        DONE35  005124        DONE37  005322        DONE4   002634
DONE5   003034        DONE6   003230        EMTAB   006366        EMTSRV  006326
END     005372        EPC     006364        FTITLE  002130        HLT     = 104006
ICOUNT  005724        KPAR0   002110        KPAR1   002112        KPAR2   002114
KPAR3   002116        KPAR4   002120        KPAR5   002122        KPAR6   002124
KPAR7   002126        KPDR0   002070        KPDR1   002072        KPDR2   002074
KPDR3   002076        KPDR4   002100        KPDR5   002102        KPDR6   002104
KPDR7   002106        KSTACK  001000        KTSTA   002026        KTVEC   002024
LOGIC   005362        MPC     005426        MPS     005433        MTIT    005376
NOP     = 000240      NRCNT   002140        NRKEYS  002142        NXTST   004330
NXTST1  004346        OK31    004450        ORDER   006526        ORDERA  006602
ORDERB  006564        PATCH1  = 000000      PATCH2  = 000000      PATCH3  = 000000
PC      =%000007      PRFLG   006316        PRINT   005732        PROCT   006106
PRSFLG  006314        PRSHRT  006054        PS      = 177776      PTEMP1  006320
PTEMP2  006322        PTEMP3  006324        P.CK    006156        P.CNT1  006232
P.CONT  006214        P.WAIT  006174        RERUNA  003300        RETRNX  005630
RETURN  005730        RET1    002426        RET21   003360        RET23   003556
RET25   003754        RET27   004152        RET31   004516        RET33   004716
RET35   005114        RET37   005312        RET4    002560        RET5    002756
RET6    003154        RWALL   005442        RWL1    005452        RWL2    005456
R0      =%000000      R1      =%000001      R2      =%000002      R3      =%000003
R4      =%000004      R5      =%000005      R6      =%000006      R7      =%000007
SAVPC   006050        SAVPSR  006052        SAVR0   006524        SCOPE   = 104400
SCOPEB  005674        SCOPEC  005632        SCOPEF  005726        SCOPEG  005706
SETUP   005504        SP      =%000006      SR      = 177570      SR0     002016
SR0H    002132        SR1     002020        SR1H    002134        SR2     002022
SR2H    002136        START   002150        STATUS  = 177776      TCSR    002012
TDBR    002014        TEMPN   006610        TESTCT  006612        TESTN   = 000015
TESTX   005534        TEST1   002256        TEST10  004062        TEST11  004346
TEST12  004550        TEST13  004750        TEST14  005146        TEST2   002460
TEST3   002656        TEST4   003056        TEST5   003252        TEST6   003462
TEST7   003662        TYPA    006462        TYPB    006504        TYPDAT  006522
TYPE    006446        UPAR0   002050        UPAR1   002052        UPAR2   002054
UPAR3   002056        UPAR4   002060        UPAR5   002062        UPAR6   002064
UPAR7   002066        UPDR0   002030        UPDR1   002032        UPDR2   002034
UPDR3   002036        UPDR4   002040        UPDR5   002042        UPDR6   002044
UPDR7   002046        USTACK  002000        XLOOP   005620        .       = 006614
```

ERRORS DETECTED:  0

IDENTIFICATION

| | |
|---|---|
| PRODUCT CODE: | MAINDEC-11-DBKTC-A-D |
| PRODUCT NAME: | MTPI/MFPI WITH MEMORY MANAGEMENT |
| DATE CREATED: | SEPTEMBER 1, 1972 |
| MAINTAINER: | DIAGNOSTIC GROUP |
| AUTHOR: | ROBERT WHITTON |

COPYRIGHT © 1972
DIGITAL EQUIPMENT CORPORATION

1,0 ABSTRACT

PROGRAM DBKTC TESTS THE MFPI AND MTPI INSTRUCTIONS WITH MEMORY MANAGEMENT ENABLED, THESE INSTRUCTIONS ARE EXECUTED IN ALL COMBINATIONS OF CURRENT MODES AND EQUAL OR LOWER HEIRARCHY PREVIOUS MODES.

2,0 REQUIREMENTS

2,1 EQUIPMENT

PDP-11/40 WITH KT11-D (MEMORY MANAGEMENT) OPTION INSTALLED

2,2 STORAGE

PROGRAM STORAGE - THE ROUTINE USES MEMORY 0-17777

3,0 LOADING AND STARTING PROCEDURE

LOAD PROGRAM INTO MEMORY USING ABS LOADER
LOAD ADDRESS 200
PRESS START,
THE PROGRAM WILL LOOP AND RING BELL ON COMPLETION.

4,0 SWITCH SETTINGS

5,0 SUBROUTINE ABSTRACTS

5,1 HLT

THE HLT (HALT) INSTRUCTION IS EXECUTED WHEN AN ERROR IS DETECTED, NOTE THAT THE HLT (HALT) INSTRUCTION TRAPS TO LOC 10 IN USER MODE, IF A HLT (HALT) INSTRUCTION IS EXECUTED IN THESE MODES THE TRAP IS TAKEN AND THE PROGRAM RETURNS TO THE HLT IN KERNEL MODE AND HALTS, NOTE: THE USER STACK POINTER IS NOT AFFECTED, FURTHER TESTING SHOULD NOT BE CONTINUED (BY PRESSING CONTINUE), THE TEST SHOULD BE RESTARTED EITHER AT THE PREVIOUS SCOPE OR AT 200,

5,2 SCOPE

THE SCOPE (EMT) SERVICE ROUTINE STORES IN R1 THE PC OF THE LAST TEST SUCCESSFULLY EXECUTED AND MAY BE USED AS AN AID IN DEBUGGING IF THE PROGRAM /BOMBS/ BECAUSE OF A HARDWARE FAILURE. A BRANCH INSTRUCTION MAY BE INSERTED AT THE SCOPE LOCATION TO THE PREVIOUS SCOPE (EMT) INSTRUCTION TO CONTINOUSLY LOOP A TEST, ADDITIONALLY THE SCOPE ROUTINE SETS ALL STACK POINTERS TO THEIR INITIAL SETTINGS (SEE SEC 8,2) AND ENTERS EACH TEST IN KERNEL MODE, PREVIOUS KERNEL MODE, ALL TESTS MAY BE RESTARTED AT THE PREVIOUS SCOPE.

6,0 ERRORS

THE TEST HALTS WHEN AN ERROR IS DETECTED AND DISPLAYS THE PC+2 OF THE HLT (HALT) INSTRUCTION IN THE ADDRESS LIGHTS.

6,1 ERROR RECOVERY

PRESS CONTINUE OR RESTART AT 200 OR PREVIOUS SCOPE.

6,2 ERROR LOOPING

TO LOOP ON AN ERROR REPLACE THE HLT INSTRUCTION WITH A BRANCH BACK TO THE PREVIOUS SCOPE. NOTE: IF THE ERROR IS INTERMITTENT THE TEST WILL DROP THROUGH THE HLT AND CONTINUE TO THE NEXT TEST.

6,3 MEMORY MANAGEMENT ABORT ERRORS

IF AN ABORT OCCURS (EXCEPT WHEN A TEST EXPECTS AN ABORT) THE PROGRAM WILL TRAP, THE TRAP SERVICE ROUTINE SAVES THE CONTENTS OF SR0 IN LOCATION SSR0T, CLEARS SR0, JUMPS TO LOCATION 252 AND HALTS. TO DETERMINE WHICH TEST CAUSED THE ABORT EITHER EXAMINE THE KERNEL STACK OR EXAMINE R1 (R1 CONTAINS THE PC OF THE FIRST INSTRUCTION IN THE TEST).

7.0 RESTRICTIONS

7.1 STARTING RESTRICTION

NONE

7.2 OPERATIONAL RESTRICTION

NONE

8.0 MISCELLANEOUS

IF THE PROGRAM HALTS IN THE TRAP/INTERRUPT VECTOR AREA (0-1000), EXAMINE REGISTER 6 (THE KERNEL STACK PTR); R6 CONTAINS THE ADDRESS WHERE THE PC OF THE INSTRUCTION THAT CAUSED THE TRAP ABORT IS STORED; SEE ALSO R1 (R1 SPECIFIES THE LAST TEST COMPLETED).

8.1 NOTE THAT THE PROGRAM TAGS EACH MFPI INSTRUCTION UNDER TEST; THE TAG DENOTES CURRENT SPACE, 'PREVIOUS' SPACE. FOR EXAMPLE:

1) $KU_{14}$:
2) UU17:

DENOTE:

1) 'CURRENT' KERNEL MODE, 'PREVIOUS' USER MODE
2) 'CURRENT' USER MODE, 'PREVIOUS' USER MODE.

NOTE ALSO THAT MEMORY MANAGEMENT IS ENABLED ONLY WHEN THE MFPI/MTPI INSTRUCTION BEING TESTED IS EXECUTED AND IS OFF AT ALL OTHER TIMES.

8,2 STACK POINTER

THE STACK POINTERS ARE INITIALLY SET TO THE FOLLOWING VALUES
KERNEL =1060
USER = 600

AND ARE RESET TO THESE VALUES AT THE START OF EACH SUBTEST (BY SCOPE).

8,3 PASS COUNT

1000(8) PASSES ARE REQUIRED FOR COMPLETION OF THIS PROGRAM, AT WHICH TIME THE BELL WILL RING AT THE TTY.

8,4 DEBUGGING TIPS

WHEN THE FAILING SUBTEST HAS BEEN ISOLATED, REPLACE THE FIRST WORD OF THE MTPI INSTRUCTION WITH A BR SELF (000777), AND START THE SUBTEST AT THE PREVIOUS SCOPE. STOP THE PROGRAM (SINGLE INSTRUCTION) AND RESTORE THE REPLACED INSTRUCTION; USING THE MAINTENANCE CARD SINGLE STEP THE FAILING INSTRUCTION THROUGH EACH MICRO STATE OBSERVING THE FLOW IN THE DATA/ADDRESS LIGHTS. THIS PRACTICE HAS BEEN FOUND TO BE SUCCESSFUL IN FINDING MOST MEMORY MANAGEMENT ERRORS.

8,5 MEMORY MANAGEMENT MEMORY MAP

THE MAPPING OF THE MEMORY MANAGEMENT REGISTERS IS DONE AT THE BEGINNING OF THE PROGRAM BEFORE ANY TESTING IS STARTED. THE USER SHOULD ACQUAINT HIMSELF WITH THE MEMORY MANAGEMENT MAP BEFORE USING THIS PROGRAM.

```
1
                                ,NLIST  SEQ
                                ,ABS
                                ,LIST   ME
                                ,TITLE  MOVV5
                        ;SEGMENTATION TEST, THIS TEST TESTS THE MTPI & MFPI INSTRUCTIONS,

                        ;GENERAL REGISTER ASSIGNMENTS
000000                          R0=%0
000001                          R1=%1
000002                          R2=%2
000003                          R3=%3
000004                          R4=%4
000005                          R5=%5
000007                          PC=%7
                        ;STACK POINTER REGISTERS
000006                          KSP=%6                  ;KERNEL STACK POINTER
000006                          USP=%6                  ;USER STACK POINTER

                        ;STATUS REGISTER BIT ASSIGNMENTS
000001                          C=1                     ;CARRY BIT
000004                          Z=4                     ;ZERO BIT
000010                          N=10                    ;NEGATIVE BIT
000340                          PRTY7=340               ;PRIORITY LEVEL 7
000200                          PRTY4=200               ;PRIORITY LEVEL 4
000000                          KM=000000               ;KERNEL MODE
140000                          UM=140000               ;USER MODE
030000                          PUM=030000              ;PREVIOUS USER MODE

                        ;VECTOR ADDRESSES
000010                          ERRVEC=10               ;ADDRESS OF ERROR VECTOR
000024                          PFVEC=24                ;ADDRESS OF POWER FAIL TRAP VECTOR
000030                          EMTVEC=30               ;ADDRESS OF EMT VECTOR
000250                          MMVEC=250               ;ADDRESS OF MNGT ERROR TRAP VECTOR

                        ;REGISTER ADDRESSES
177776                          PSW=177776              ;ADDRESS OF STATUS REGISTER
177774                          SLR=177774              ;ADDRESS OF STACK LIMIT REGISTER
177560                          TKS=177560              ;ADDRESS OF KEYBOARD CSR
177562                          TKB=177562              ;ADDRESS OF KEYBOARD BUFFER
177564                          TPS=177564              ;ADDRESS OF TELEPRINTER CSR
177566                          TPB=177566              ;ADDRESS OF TELEPRINTER BUFFER
177570                          DISPLAY=177570          ;ADDRESS OF CONSOL DISPLAY REGISTER
```

```
                        ;INITIAL STACK POINTER SETTINGS
001060                          KPTR=1060               ;BOTTOM OF KERNEL STACK
000600                          UPTR=600                ;USER STACK SETTING
                        ;*****************************NOTE*********************************
                        ;THE KERNEL & USER STACK POINTER ARE AT PHYSICAL 1060 & 0600
                        ;******************************************************************

                        ;MNGTMENTATION REGISTER ADDRESS ASSIGNMENTS
177572                          SR0=177572              ;ADDRESS OF SEGMENTATION REGISTER SR0
177574                          SR1=177574              ;    "    "       "           "    SR1
177576                          SR2=177576              ;    "    "       "           "    SR2

177600                          UPDR0=177600            ;USER PDR'S
177602                          UPDR1=177602
177604                          UPDR2=177604
177606                          UPDR3=177606
177610                          UPDR4=177610
177612                          UPDR5=177612
177614                          UPDR6=177614
177616                          UPDR7=177616

177640                          UPAR0=177640            ;USER PAR'S
177642                          UPAR1=177642
177644                          UPAR2=177644
177646                          UPAR3=177646
177650                          UPAR4=177650
177652                          UPAR5=177652
177654                          UPAR6=177654
177656                          UPAR7=177656


172300                          KPDR0=172300            ;KERNEL PDR'S
172302                          KPDR1=172302
172304                          KPDR2=172304
172306                          KPDR3=172306
172310                          KPDR4=172310
172312                          KPDR5=172312
172314                          KPDR6=172314
172316                          KPDR7=172316

172340                          KPAR0=172340            ;KERNEL PAR'S
172342                          KPAR1=172342
172344                          KPAR2=172344
172346                          KPAR3=172346
172350                          KPAR4=172350
172352                          KPAR5=172352
172354                          KPAR6=172354
172356                          KPAR7=172356
```

```
                                ;ACCESS CONTROL FIELD DEFINITIONS (IN PDR)
            000006                      RW=6                        ;READ & WRITE

                                ;INSTUCTION EQUATES
            000000                      HLT=HALT
            104000                      SCOPE=EMT                   ;SCOPE IS AN EMT TRAP INST.


            000000                      .=0
                                        .REPT   100
                                        .+2
                                        HALT
                                        .ENDR
  000000    000002                      .+2
  000002    000000                      HALT
  000004    000006                      .+2
  000006    000000                      HALT
  000010    000012                      .+2
  000012    000000                      HALT
  000014    000016                      .+2
  000016    000000                      HALT
  000020    000022                      .+2
  000022    000000                      HALT
  000024    000026                      .+2
  000026    000000                      HALT
  000030    000032                      .+2
  000032    000000                      HALT
  000034    000036                      .+2
  000036    000000                      HALT
  000040    000042                      .+2
  000042    000000                      HALT
  000044    000046                      .+2
  000046    000000                      HALT
  000050    000052                      .+2
  000052    000000                      HALT
  000054    000056                      .+2
  000056    000000                      HALT
  000060    000062                      .+2
  000062    000000                      HALT
  000064    000066                      .+2
  000066    000000                      HALT
  000070    000072                      .+2
  000072    000000                      HALT
  000074    000076                      .+2
  000076    000000                      HALT
  000100    000102                      .+2
  000102    000000                      HALT
  000104    000106                      .+2
  000106    000000                      HALT
  000110    000112                      .+2
  000112    000030                      HALT
  000114    000116                      .+2
  000116    000000                      HALT
```

```
000120  000122                    .+2
000122  000000                    HALT
000124  000126                    .+2
000126  000000                    HALT
000130  000132                    .+2
000132  000000                    HALT
000134  000136                    .+2
000136  000000                    HALT
000140  000142                    .+2
000142  000000                    HALT
000144  000146                    .+2
000146  000000                    HALT
000150  000152                    .+2
000152  000000                    HALT
000154  000156                    .+2
000156  000000                    HALT
000160  000162                    .+2
000162  000000                    HALT
000164  000166                    .+2
000166  000000                    HALT
000170  000172                    .+2
000172  000000                    HALT
000174  000176                    .+2
000176  000000                    HALT
000200  000202                    .+2
000202  000000                    HALT
000204  000206                    .+2
000206  000000                    HALT
000210  000212                    .+2
000212  000000                    HALT
000214  000216                    .+2
000216  000000                    HALT
000220  000222                    .+2
000222  000000                    HALT
000224  000226                    .+2
000226  000000                    HALT
000230  000232                    .+2
000232  000000                    HALT
000234  000236                    .+2
000236  000000                    HALT
000240  000242                    .+2
000242  000000                    HALT
000244  000246                    .+2
000246  000000                    HALT
000250  000252                    .+2
000252  000000                    HALT
000254  000256                    .+2
000256  000000                    HALT
000260  000262                    .+2
000262  000000                    HALT
000264  000266                    .+2
000266  000000                    HALT
000270  000272                    .+2
000272  000000                    HALT
```

```
000274  000276                    .+2
000276  000000                    HALT
000300  000302                    .+2
000302  000000                    HALT
000304  000306                    .+2
000306  000000                    HALT
000310  000312                    .+2
000312  000000                    HALT
000314  000316                    .+2
000316  000000                    HALT
000320  000322                    .+2
000322  000000                    HALT
000324  000326                    .+2
000326  000000                    HALT
000330  000332                    .+2
000332  000000                    HALT
000334  000336                    .+2
000336  000000                    HALT
000340  000342                    .+2
000342  000000                    HALT
000344  000346                    .+2
000346  000000                    HALT
000350  000352                    .+2
000352  000000                    HALT
000354  000356                    .+2
000356  000000                    HALT
000360  000362                    .+2
000362  000000                    HALT
000364  000366                    .+2
000366  000000                    HALT
000370  000372                    .+2
000372  000000                    HALT
000374  000376                    .+2
000376  000000                    HALT

        000010                    .=ERRVEC
000010  000400                    .WORD   SHLT            ;SET USER HALT TRAP
        000030                    .=EMTVEC
000030  000432                    .WORD   SCOPEA          ;SET SCOPE (EMT) TRAP VECTOR
        000250                    .=MMVEC
000250  000462                    .WORD   MMERR           ;SET SEG. ERROR TRAP VECTOR


                                  .NLIST  MC
        000200                    .=200
000200  000167  000664            JMP     START           ;GO START TEST
```

```
                000400                          .=400

                                        ;USER HALT TRAP SERVICE ROUTINE.
000400  162716  000002          SHLT:   SUB     #2,(KSP)        ;ADJUST PC
000404  005776  000000                  TST     @(KSP)          ;CHECK IF HLT CAUSED TRAP
000410  001404                          BEQ     SHLTA
000412  062716  000002                  ADD     #2,(KSP)        ;RESTORE PC
000416  000167  177364                  JMP     6               ;GO HALT AT 6
000422  042766  140000  000002  SHLTA:  BIC     #UM,2(KSP)      ;KERNEL MODE ON RETURN
000430  000002                          RTI                     ;RETURN


                                        ;SCOPE (EMT) SERVICE ROUTINE
000432  011601                  SCOPEA: MOV     (KSP),R1        ;SAVE RETURN ADDRESS IN R1
000434  012706  001060                  MOV     #KPTR,KSP       ;SET KERNEL STACK PTR
000440  005046                          CLR     -(KSP)
000442  010146                          MOV     R1,-(KSP)
000444  012746  000600                  MOV     #UPTR,-(KSP)
000450  012737  030000  177776          MOV     #PUM,@#PSW      ;PREVIOUS USER MODE
000456  006606                          MTPI    USP             ;SET USER STACK PTR
000460  000002                          RTI                     ;RETURN & START NEXT TEST

                                        ;SEGMENTATION ERROR SERVICE
000462  013767  177572  000312  MMERR:  MOV     @#SR0,SSR0T     ;SAVE SR0
000470  005037  177572                  CLR     @#SR0
000474  000137  000252                  JMP     @#MMVEC+2

                001000                  .=1000
                                        ;TAGS
001000  000000                  ICNT:   0                       ;CONTAINS PASS COUNT
001002  000000                  SSR0T:  0                       ;CONTAINS SSR0 CONTENTS ON ERROR
001004  000000                  TEMP:   0
                001014                  .=.+6
```

```
                001070                          .=1070
                                        ;START SEGMENTATION TEST-S5
001070  000240                  START:  NOP
001072  005067  177702                  CLR     ICNT                    ;CLEAR PASS COUNT
001076  005037  177776          BEGIN:  CLR     @#PSW           ;KERNEL MODE!!!,PREV KERNEL MOD
001102  012706  001060                  MOV     #KPTR,KSP       ;SET KERNEL STACK PTR
001106  104000                          SCOPE                   ;SCOPE SETS UP ALL STACK PTRS
001110  012737  000462  000250          MOV     #MMERR,@#MMVEC
001116  005037  000252                  CLR     @#MMVEC+2       ;KERNEL MODE ON SEG ABORT
001122  000240                          NOP                     ;CLEAR MEMORY MANAGEMENT REGISTERS
001124  005037  177572                  CLR     @#SR0
001130  012702  177600                  MOV     #UPDR0,R2
001134  012703  000010                  MOV     #10,R3
001140  005022                          CLR     (R2)+
001142  077302                          SOB     R3,.-2
001144  012702  177640                  MOV     #UPAR0,R2
001150  012703  000010                  MOV     #10,R3
001154  005022                          CLR     (R2)+
001156  077302                          SOB     R3,.-2
001160  012702  172300                  MOV     #KPDR0,R2
001164  012703  000010                  MOV     #10,R3
001170  005022                          CLR     (R2)+
001172  077302                          SOB     R3,.-2
001174  012702  172340                  MOV     #KPAR0,R2
001200  012703  000010                  MOV     #10,R3
001204  005022                          CLR     (R2)+
001206  077302                          SOB     R3,.-2
001210  012767  073006  171062          MOV     #73006,KPDR0    ;RW,UP 167 BLOCKS
001216  012767  000006  171070          MOV     #6,KPDR6        ;RW,UP 1 BLOCK
001224  012767  077406  171064          MOV     #77406,KPDR7    ;RW,UP 200 BLOCKS
001232  012767  000006  176350          MOV     #6,UPDR4        ;RW,UP 1 BLOCK
001240  012767  000006  176344          MOV     #6,UPDR5        ;RW,UP 1 BLOCK
001246  005067  171066                  CLR     KPAR0           ;VA=PA=0000-12077
001252  012767  000167  171074          MOV     #167,KPAR6      ;VA=140000-140077;PA=16700-16777
001260  012767  007600  171070          MOV     #7600,KPAR7     ;VA=160000-177776,PA=760000-777776
001266  012767  000173  176354          MOV     #173,UPAR4      ;VA=100000-100077/PA=17300-17377
001274  012767  000172  176350          MOV     #172,UPAR5      ;VA=120000-120077/PA=17400-17277
```

```
                                        ;TESTS KK0-KK16 TEST THE MTPI INSTRUCTION KERNEL MODE, PREV KERNEL MODE,
                                        ;*******************************
            016600                              VIRT=16600              ;KERNEL VIRTUAL ADDRESS FOR THESE TESTS
            016600                              PHYS=16600              ;CORRESPONDING KERNEL PHYSICAL ADDRESS
                                        ;*******************************

                                        ;TEST THAT MTPI CAN LOAD A GENERAL REGISTER (R2)
001302  005016                                  CLR     (KSP)           ;PUT 0 ON KERNEL STACK
001304  012702  177777                          MOV     #-1,R2          ;PRESET REGISTER
001310  005237  177572                          INC     @#SR0           ;ENABLE KT11

001314  006602                          KK0:    MTPI    R2              ;R2=(KSP)+
001316  016703  176454                          MOV     PSW,R3          ;SAVE STATUS RESULT
001322  005037  177572                          CLR     @#SR0           ;DISABLE KT11
001326  022706  001062                          CMP     #KPTR+2,KSP     ;CHECK THAT STACK POPPED
001332  001401                                  BEQ     .+4
001334  000000                                  HLT                     ;ERROR!  INCORRECT STACK PTR
001336  122703  000004                          CMPB    #Z,R3   ;CHECK STATUS RESULT
001342  001401                                  BEQ     .+4
001344  000000                                  HLT                     ;ERROR! INCORRECT STATUS RESULT
001346  005702                                  TST     R2              ;CHECK RESULT
001350  001401                                  BEQ     .+4
001352  000000                                  HLT                     ;ERROR!  INCORRECT RESULT
001354  104000                                  SCOPE


                                        ;TEST THAT MTPI CAN LOAD KERNEL ADDRESS (VIRT)
                                        ;DM=1

001356  005016                                  CLR     (KSP)           ;PUT 0 ON KERNEL STACK
001360  012702  016600                          MOV     #VIRT,R2                ;R2=VIRT ADDRESS
001364  012737  177777  016600                  MOV     #-1,@#PHYS      ;PRESET DATA
001372  005237  177572                          INC     @#SR0           ;ENABLE KT11

001376  000277                                  SCC
001400  006612                          KK1:    MTPI    (R2)            ;VIRT=(KSP)+
001402  016703  176370                          MOV     PSW,R3          ;SAVE STATUS RESULT
001406  005037  177572                          CLR     @#SR0           ;DISABLE KT11
001412  022706  001062                          CMP     #KPTR+2,KSP     ;CHECK THAT STACK POPPED
001416  001401                                  BEQ     .+4
001420  000000                                  HLT                     ;ERROR! INCORRECT STACK PTR
001422  122703  000005                          CMPB    #Z+C,R3         ;CHECK STATUS RESULT
001426  001401                                  BEQ     .+4
001430  000000                                  HLT                     ;ERROR! INCORRECT STATUS RESULT
001432  005737  016600                          TST     @#PHYS          ;CHECK RESULT
001436  001401                                  BEQ     .+4
001440  000000                                  HLT                     ;ERROR! INCORRECT RESULT
001442  104000                                  SCOPE

                                        ;DM=2
001444  012737  000000  177776                  MOV     #0,@#PSW        ;KERNEL MODE!!!,PREV KERNEL MODE
001452  012716  177777                          MOV     #-1,(KSP)       ;PUT #-1 ON KERNEL STACK
001456  012702  016600                          MOV     #VIRT,R2        ;R2=VIRT ADDRESS
```

```
001462  005037  016600                CLR     @#PHYS          ;PRESET DATA
001466  005237  177572                INC     @#SR0           ;ENABLE KT11

001472  006622                KK2:    MTPI    (R2)+           ;VIRT=(KSP)+
001474  005037  177572                CLR     @#SR0           ;DISABLE KT11
001500  005237  016600                INC     @#PHYS          ;CHECK RESULT
001504  001401                        BEQ     .+4
001506  000000                        HLT                     ;ERROR! INCORRECT RESULT
001510  022702  016602                CMP     #VIRT+2,R2      ;CHECK AUTO-INCREMENT
001514  001401                        BEQ     .+4
001516  000000                        HLT                     ;ERROR! AUTO-INCREMENT FAILED
001520  005067  176252                CLR     PSW
001524  104000                        SCOPE

                              ;DM=3
001526  012716  177777                MOV     #-1,(KSP)       ;PUT #-1 ON KERNEL STACK
001532  012702  001004                MOV     #TEMP,R2        ;LOAD INDIRECT ADDRESS
001536  012737  016604  001004        MOV     #VIRT+4,@#TEMP  ;LOAD ADDRESS
001544  005037  016604                CLR     @#PHYS+4        ;PRESET DATA
001550  005237  177572                INC     @#SR0           ;ENABLE KT11

001554  006632                KK3:    MTPI    @(R2)+          ;VIRT+4=(KSP)+
001556  005037  177572                CLR     @#SR0           ;DISABLE KT11
001562  005237  016604                INC     @#PHYS+4                ;CHECK RESULT
001566  001401                        BEQ     .+4
001570  000000                        HLT                     ;ERROR! INCORRECT RESULT
001572  104000                        SCOPE

                              ;DM=4
001574  005016                        CLR     (KSP)           ;PUT 0 ON KERNEL STACK
001576  012704  016602                MOV     #VIRT+2,R4      ;LOAD ADDRESS
001602  012737  177777  016600        MOV     #-1,@#PHYS      ;PRESET DATA
001610  005237  177572                INC     @#SR0           ;ENABLE KT11

001614  006644                KK4:    MTPI    -(R4)           ;VIRT=-(KSP)+
001616  005037  177572                CLR     @#SR0           ;DISABLE KT11
001622  022704  016600                CMP     #VIRT,R4                ;CHECK AUTO-DECREMENT
001626  001401                        BEQ     .+4
001630  000000                        HLT                     ;ERROR! AUTO-DECREMENT FAILED
001632  005737  016600                TST     @#PHYS          ;CHECK RESULT
001636  001401                        BEQ     .+4
001640  000000                        HLT                     ;ERROR! INCORRECT RESULT
001642  104000                        SCOPE

                              ;DM=5
001644  012737  000000  177776        MOV     #0,@#PSW        ;KERNEL MODE!!!,PREV KERNEL MODE
001652  012716  177777                MOV     #-1,(KSP)       ;PUT #-1 ON KERNEL STACK
001656  012702  001010                MOV     #TEMP+4,R2      ;LOAD INDIRECT ADDRESS
001662  012767  016600  177116        MOV     #VIRT,TEMP+2    ;LOAD ADDRESS
001670  005037  016600                CLR     @#PHYS          ;PRESET DATA
001674  005237  177572                INC     @#SR0           ;ENABLE KT11

001700  006652                KK5:    MTPI    @-(R2)          ;VIRT=-(KSP)+
001702  005037  177572                CLR     @#SR0           ;DISABLE KT11
001706  005237  016600                INC     @#PHYS          ;CHECK RESULT
```

```
001712  001401                          BEQ     .+4
001714  000000                          HLT                     ;ERROR! INCORRECT RESULT
001716  005067  176054                  CLR     PSW
001722  104000                          SCOPE

                                ;DM=6
001724  012737  000000  177776          MOV     #0,@#PSW        ;KERNEL MODE!!!,PREV KERNEL MODE
001732  005016                          CLR     (KSP)           ;PUT 0 ON KERNEL STACK
001734  012702  000002                  MOV     #2,R2           ;LOAD INDEX REGISTER
001740  012767  177777  014634          MOV     #-1,PHYS+2      ;PRESET DATA
001746  005237  177572                  INC     @#SR0           ;ENABLE KT11

001752  006662  016600          KK6:    MTPI    VIRT(R2)        ;VIRT+2-(KSP)+
001756  016700  176014                  MOV     PSW,R0          ;SAVE STATUS RESULT
001762  005037  177572                  CLR     @#SR0           ;DISABLE KT11
001766  022706  001062                  CMP     #KPTR+2,KSP     ;CHECK THAT STACK POINTER POPPED
001772  001401                          BEQ     .+4
001774  000000                          HLT                     ;ERROR! INCORRECT STACK PTR
001776  122700  000004                  CMPB    #2,R0   ;CHECK STATUS RESULT
002002  001401                          BEQ     .+4
002004  000000                          HLT                     ;ERROR! INCORRECT STATUS RESULT
002006  005737  016602                  TST     @#PHYS+2                ;CHECK RESULT
002012  001401                          BEQ     .+4
002014  000000                          HLT                     ;ERROR! INCORRECT RESULT
002016  104000                          SCOPE

                                ;DM=7
002020  012716  177777                  MOV     #-1,(KSP)       ;PUT #-1 ON KERNEL STACK
002024  012702  000002                  MOV     #2,R2           ;LOAD INDEX REGISTER
002030  012737  016600  001010          MOV     #VIRT,@#TEMP+4  ;LOAD ADDRESS
002036  005037  016600                  CLR     @#PHYS          ;PRESET DATA
002042  005237  177572                  INC     @#SR0           ;ENABLE KT11

002046  006672  001006          KK7:    MTPI    @TEMP+2(R2)     ;VIRT+(KSP)+
002052  005037  177572                  CLR     @#SR0           ;DISABLE KT11
002056  005237  016600                  INC     @#PHYS          ;CHECK RESULT
002062  001401                          BEQ     .+4
002064  000000                          HLT                     ;ERROR! INCORRECT RESULT
002066  104000                          SCOPE

                                ;TEST THAT MTPI CAN LOAD KERNEL
                                ;DM=1,PC
002070  012716  000403          KK10B:  MOV     #403,(KSP)      ;PUT BR ,+10 INST AS DATA ON STACK
002074  005037  002106                  CLR     @#KK10A ;PUT HALT FOLLOWING MTPI
002100  005237  177572                  INC     @#SR0           ;ENABLE KT11

002104  006617                  KK10:   MTPI    (PC)            ;KK10A+(KSP)+
002106  000000                  KK10A:  HALT                    ;ERROR!  MTPI DID NOT POP BR .+10
                                                                ;INTO KK10A
002110  005037  177572                  CLR     @#SR0           ;DISABLE KT11
002114  000765                          BR      KK10B           ;LOOP TEST IF ERROR
002116  005037  177572                  CLR     @#SR0           ;DISABLE KT11
002122  104000                          SCOPE
```

```
                                        ;DM=2,PC
002124  012716  177777                          MOV     #-1,(KSP)       ;PUT #-1 ON KERNEL STACK
002130  005067  000006                          CLR     KK11A
002134  005237  177572                          INC     @#SR0           ;ENABLE KT11

002140  006627                          KK11:   MTPI    (PC)+           ;(PC)+-(KSP)+
002142  000000                          KK11A:  0
002144  005037  177572                          CLR     @#SR0           ;DISABLE KT11
002150  005267  177766                          INC     .-6             ;CHECK THAT DATA POPPED TO
002154  001401                          BEQ     .+4             ;CORRECT ADDRESS
002156  000000                          HLT                     ;ERROR!
002160  104000                          SCOPE

                                        ;DM=3,PC
002162  012737  000000  177776          MOV     #0,@#PSW        ;KERNEL MODE!!!,PREV KERNEL MODE
002170  012716  177777                  MOV     #-1,(KSP)       ;PUT #-1 ON KERNEL STACK
002174  005037  016600                  CLR     @#PHYS
002200  005237  177572                  INC     @#SR0           ;ENABLE KT11

002204  006637  016600          KK12:   MTPI    @#VIRT          ;VIRT-(KSP)+
002210  016700  175562                  MOV     PSW,R0          ;SAVE STATUS RESULT
002214  005037  177572                  CLR     @#SR0           ;DISABLE KT11
002220  122700  000010                  CMPB    #N,R0   ;CHECK STATUS RESULT
002224  001401                          BEQ     .+4
002226  000000                          HLT                     ;ERROR!  INCORRECT STATUS RESULT
002230  005267  014344                  INC     PHYS            ;CHECK RESULT
002234  001401                          BEQ     .+4
002236  000000                          HLT                     ;ERROR!  INCORRECT RESULT
002240  005067  175532                  CLR     PSW
002244  104000                          SCOPE

                                        ;DM=4,PC
002246  012737  000000  177776          MOV     #0,@#PSW        ;KERNEL MODE!!!,PREV KERNEL MODE
002254  012716  000401                  MOV     #401,(KSP)      ;PUT BR .+4 ON STACK AS DATA
002260  016702  000004                  MOV     KK13,R2 ;SAVE MTPI INSTRUCTION
002264  005237  177572                  INC     @#SR0           ;ENABLE KT11

002270  006647                  KK13:   MTPI    -(PC)           ;-(PC)+-(KSP)+
002272  000000                          HALT                    ;ERROR!  MTPI -(PC) FAILED
002274  005037  177572                  CLR     @#SR0           ;DISABLE KT11
002300  010267  177764                  MOV     R2,KK13 ;RESTORE INSTRUCTION
002304  104000                          SCOPE

                                        ;DM=6,PC
002306  005016                          CLR     (KSP)           ;PUT 0 ON KERNEL STACK
002310  012767  177777  014266          MOV     #-1,PHYS+4
002316  005237  177572                  INC     @#SR0           ;ENABLE KT11

002322  000277                          SCC
002324  006667  014254          KK14:   MTPI    VIRT+4          ;VIRT+4-(KSP)+
002330  016703  175442                  MOV     PSW,R3          ;SAVE STATUS RESULT
002334  005037  177572                  CLR     @#SR0           ;DISABLE KT11
002340  022706  001062                  CMP     #KPTR+2,KSP     ;CHECK THAT STACK PTR POPPED
002344  001401                          BEQ     .+4
```

```
002346  000000                        HLT                             ;ERROR!  INCORRECT STACK PTR
002350  122703  000005                CMPB    #Z+C,R3 ;CHECK STATUS RESULT
002354  001401                        BEQ     .+4
002356  000000                        HLT                             ;ERROR!  INCORRECT STATUS RESULT
002360  005737  016604                TST     @#PHYS+4                        ;CHECK RESULT
002364  001401                        BEQ     .+4
002366  000000                        HLT                             ;ERROR!  INCORRECT RESULT

002370  104000                        SCOPE
                                ;DM=7,PC

002372  005037  177776                CLR     @#PSW                   ;KERNEL MODE!!!,PREV KERNEL MODE!!
002376  012716  177777                MOV     #-1,(KSP)               ;PUT #-1 ON KERNEL STACK
002402  012737  016604  001004        MOV     #VIRT+4,@#TEMP          ;LOAD ADDRESS
002410  005037  016604                CLR     @#PHYS+4                        ;PRESET DATA
002414  005237  177572                INC     @#SR0                   ;ENABLE KT11


002420  006677  176360        KK15:   MTPI    @TEMP                   ;VIRT+4+(KSP)+
002424  005037  177572                CLR     @#SR0                   ;DISABLE KT11
002430  005237  016604                INC     @#PHYS+4                        ;CHECK RESULT
002434  001401                        BEQ     .+4
002436  000000                        HLT                             ;ERROR INCORRECT RESULT
002440  104000                        SCOPE


                                ;CHECK THAT MTPI CAN SET STACK PTR
002442  012737  000000  177776        MOV     #0,@#PSW                ;KERNEL MODE!!!,PREV KERNEL MODE
002450  005016                        CLR     (KSP)                   ;PUT 0 ON KERNEL STACK
002452  005237  177572                INC     @#SR0                   ;ENABLE KT11

002456  006606                KK16:   MTPI    KSP                     ;KSP+(KSP)+
002460  005037  177572                CLR     @#SR0                   ;DISABLE KT11
002464  005706                        TST     KSP                     ;CHECK STACK PTR
002466  001401                        BEQ     .+4
002470  000000                        HLT                             ;ERROR!
002472  012706  001060                MOV     #KPTR,KSP               ;SET KERNEL STACK PTR
002476  104000                        SCOPE

                                ;TESTS KU0-KU16 TEST THE MTPI INSTRUCTION KERNEL MODE, PREV USER MODE,
                                ;TEST THAT MTPI CAN LOAD A GENERAL REGISTER (R2)
002500  012737  030340  177776        MOV     #PUM+PRTY7,@#PSW                ;KERNEL MODE!!!,PREV USER MODE!!
002506  005016                        CLR     (KSP)                   ;PUT 0 ON KERNEL STACK
002510  012702  177777                MOV     #-1,R2                  ;PRESET REGISTER
002514  005237  177572                INC     @#SR0                   ;ENABLE KT11

002520  006602                KU0:    MTPI    R2                      ;R2+(KSP)+
002522  016703  175250                MOV     PSW,R3                  ;SAVE STATUS RESULT
002526  005037  177572                CLR     @#SR0                   ;DISABLE KT11
002532  022706  001062                CMP     #KPTR+2,KSP             ;CHECK THAT STACK POPPED
002536  001401                        BEQ     .+4
002540  000000                        HLT                             ;ERROR!  INCORRECT STACK PTR
002542  122703  000344                CMPB    #PRTY7+Z,R3             ;CHECK STATUS RESULT
002546  001401                        BEQ     .+4
```

```
002550  000000                          HLT                     ;ERROR! INCORRECT STATUS RESULT
002552  005702                          TST     R2              ;CHECK RESULT
002554  001401                          BEQ     .+4
002556  000000                          HLT                     ;ERROR!  INCORRECT RESULT
002560  104000                          SCOPE
                                ;TEST MFPI INSTRUCUTION KERNEL MODE PREVIOUS KERNEL MODE.
                                ;TEST THAT MFPI CAN GET DATA FROM A GENRAL REGISTER (R3)
002562  012767  000340  175206          MOV     #PRTY7,PSW      ;KERNEL MODE!!!,PREV KERNEL MODE!!
002570  005066  177776                  CLR     -2(KSP)
002574  012703  177777                  MOV     #-1,R3          ;PRESET GENERAL REGISTER
002600  005237  177572                  INC     @#$R0           ;ENABLE MEMORY MANAGEMENT
002604  006503                  KKF0:   MFPI    R3              ;-(KSP)-R3
002606  016702  175164                  MOV     PSW,R2          ;SAVE CC'S
002612  005037  177572                  CLR     @#$R0           ;DISABLE MEMORY MANAGEMENT
002616  122702  000350                  CMPB    #PRTY7+N,R2     ;CHECK CC'S
002622  001401                          BEQ     .+4
002624  000000                          HLT                     ;ERROR! INCORRECT CC'S AFTER MFPI
002626  022706  001056                  CMP     #KPTR-2,KSP     ;CHECK THAT STACK WAS PUSHED
002632  001401                          BEQ     .+4
002634  000000                          HLT                     ;ERROR! INCORRECT STACK PTR
002636  005216                          INC     (KSP)           ;CHECK RESULT
002640  001401                          BEQ     .+4
002642  000000                          HLT                     ;ERROR! INCORRECT RESULT
002644  104000                          SCOPE

                                ;TEST THAT MFPI CAN GET DATA FROM A KERNEL VIRTUAL ADDRESS
                                ;DM=1
002646  005067  175124                  CLR     PSW             ;KERNEL MODE!!!,PREV KERNEL MODE!!
002652  005066  177776                  CLR     -2(KSP)
002656  012702  016600                  MOV     #VIRT,R2        ;R2=VIRTUAL ADDRESS
002662  012737  177777  016600          MOV     #-1,@#PHYS      ;PRESET PHYSICAL ADDRESS
002670  005237  177572                  INC     @#$R0           ;ENABLE MEMORY MANAGEMENT
002674  000277                          SCC                     ;PRESET CC'S
002676  006512                  KKF1:   MFPI    (R2)            ;-(KSP)-(R2)
002700  016703  175072                  MOV     PSW,R3          ;SAVE CC'S
002704  005037  177572                  CLR     @#$R0           ;DISABLE MEMORY MANAGEMENT
002710  122703  000011                  CMPB    #N+C,R3         ;CHECK CC'S
002714  001401                          BEQ     .+4
002716  000000                          HLT                     ;ERROR! INCORRECT CC'S
002720  022706  001056                  CMP     #KPTR-2,KSP     ;CHECK THAT STACK WAS PUSHED
002724  001401                          BEQ     .+4
002726  000000                          HLT                     ;ERROR! INCORRECT STACK PTR
002730  005216                          INC     (KSP)           ;CHECK RESULT
002732  001401                          BEQ     .+4
002734  000000                          HLT                     ;ERROR! INCORRECT RESULT
002736  104000                          SCOPE

                                ;DM=2
002740  012767  000000  175030          MOV     #0,PSW          ;KERNEL MODE!!!,PREV KERNEL MODE!!
002746  012766  177777  177776          MOV     #-1,-2(KSP)
002754  012702  016600                  MOV     #VIRT,R2        ;R2=VIRTUAL ADDRESS
002760  005037  016600                  CLR     @#PHYS          ;PRESET PHYSICAL ADDRESS
002764  005237  177572                  INC     @#$R0           ;ENABLE MEMORY MANAGEMENT
002770  006522                  KKF2:   MFPI    (R2)+           ;-(KSP)-VIRT
```

```
002772  005037  177572                CLR     @#SR0           ;DISABLE MEMORY MANAGEMENT
002776  005716                        TST     (KSP)           ;CHECK RESULT
003000  001401                        BEQ     .+4
003002  000000                        HLT                     ;ERROR! INCORRECT RESULT ON STACK
003004  022702  016602                CMP     #VIRT+2,R2      ;CHECK AUTO INCREMENT
003010  001401                        BEQ     .+4
003012  000000                        HLT                     ;ERROR! AUTO INCREMENT FAILED
003014  005067  174756                CLR     PSW
003020  104000                        SCOPE

                                ;DM=3
003022  005067  174750                CLR     PSW             ;KERNEL MODE!!!,PREV KERNEL MODE!!
003026  005066  177776                CLR     -2(KSP)
003032  012702  001004                MOV     #TEMP,R2        ;LOAD INDIRECT ADDRESS
003036  012737  016602  001004        MOV     #VIRT+2,@#TEMP  ;LOAD ADDRESS
003044  012737  177777  016602        MOV     #-1,@#PHYS+2    ;PRESET DATA
003052  005237  177572                INC     @#SR0           ;ENABLE MEMORY MANAGEMENT
003056  006532                KKF3:   MFPI    @(R2)+          ;-(KSP)=VIRT+2
003060  005037  177572                CLR     @#SR0           ;DISABLE MEMORY MANAGEMENT
003064  005216                        INC     (KSP)           ;CHECK RESULT
003066  001401                        BEQ     .+4
003070  000000                        HLT                     ;ERROR! INCORRECT RESULT
003072  104000                        SCOPE

                                ;DM=4
003074  005067  174676                CLR     PSW             ;KERNEL MODE!!!,PREV KERNEL MODE!!
003100  012766  177777  177776        MOV     #-1,-2(KSP)
003106  012704  016602                MOV     #VIRT+2,R4      ;R4=VIRTUAL ADDRESS+2
003112  005037  016600                CLR     @#PHYS          ;PRESET PHYSICAL ADDRESS DATA
003116  005237  177572                INC     @#SR0           ;ENABLE MEMORY MANAGEMENT
003122  006544                KKF4:   MFPI    -(R4)           ;-(KSP)=VIRT
003124  005037  177572                CLR     @#SR0           ;DISABLE MEMORY MANAGEMENT
003130  022704  016600                CMP     #VIRT,R4        ;CHECK AUTO-DECREMENT
003134  001401                        BEQ     .+4
003136  000000                        HLT                     ;ERROR! AUTO-DECREMENT FAILED
003140  005716                        TST     (KSP)           ;CHECK RESULT
003142  001401                        BEQ     .+4
003144  000000                        HLT                     ;ERROR! INCORRECT RESULT
003146  104000                        SCOPE

                                ;DM=5
003150  012767  000000  174620        MOV     #0,PSW          ;KERNEL MODE!!!,PREV KERNEL MODE!!
003156  005066  177776                CLR     -2(KSP)
003162  012700  001006                MOV     #TEMP+2,R0      ;R1=INDIRECT ADDRESS
003166  012737  016604  001004        MOV     #VIRT+4,@#TEMP  ;LOAD ADDRESS
003174  012737  177777  016604        MOV     #-1,@#PHYS+4    ;PRESET PHYSICAL ADDRESS DATA
003202  005237  177572                INC     @#SR0           ;ENABLE MEMORY MANAGEMENT
003206  006550                KKF5:   MFPI    @-(R0)          ;-(KSP)=VIRT+4
003210  005037  177572                CLR     @#SR0           ;DISABLE MEMORY MANAGEMENT
003214  005216                        INC     (KSP)           ;CHECK RESULT
003216  001401                        BEQ     .+4
003220  000000                        HLT
003222  005067  174550                CLR     PSW
003226  104000                        SCOPE
```

```
                                        ;DM=6
003230  012767  000000  174540          MOV     #0,PSW          ;KERNEL MODE!!!,PREV KERNEL MODE!!
003236  012766  177777  177776          MOV     #-1,-2(KSP)
003244  012702  000002                  MOV     #2,R2           ;LOAD INDEX REGISTER
003250  005037  016602                  CLR     @#PHYS+2        ;PRESET PHYSICAL ADDRESS DATA
003254  005237  177572                  INC     @#SR0           ;ENABLE MEMORY MANAGEMENT
003260  006562  016600          KKF6:   MFPI    VIRT(R2)        ;-(KSP)←VIRT+2
003264  005037  177572                  CLR     @#SR0           ;DISABLE MEMORY MANAGEMENT
003270  022706  001056                  CMP     #KPTR-2,KSP     ;CHECK STACK PTR
003274  001401                          BEQ     .+4
003276  000000                          HLT                     ;ERROR! INCORRECT STACK PTR
003300  005716                          TST     (KSP)           ;CHECK RESULT
003302  001401                          BEQ     .+4
003304  000000                          HLT                     ;ERROR! INCORRECT RESULT
003306  005067  174464                  CLR     PSW
003312  104000                          SCOPE

                                        ;DM=7
003314  005067  174456                  CLR     PSW             ;KERNEL MODE!!!,PREV KERNEL MODE!!
003320  005066  177776                  CLR     -2(KSP)
003324  012702  177774                  MOV     #-4,R2          ;LOAD INDEX REGISTER
003330  012737  016600  001004          MOV     #VIRT,@#TEMP    ;LOAD ADDRESS
003336  012737  177777  016600          MOV     #-1,@#PHYS      ;CLEAR PHYSICAL ADDRESS DATA
003344  005237  177572                  INC     @#SR0           ;ENABLE MEMORY MANAGEMENT
003350  006572  001010          KKF7:   MFPI    @TEMP+4(R2)     ;-(KSP)←VIRT
003354  005037  177572                  CLR     @#SR0           ;DISABLE MEMORY MANAGEMENT
003360  005216                          INC     (KSP)           ;CHECK RESULT
003362  001401                          BEQ     .+4
003364  000000                          HLT                     ;ERROR! INCORRECT RESULT
003366  104000                          SCOPE

                                        ;TEST THAT MFPI OPERATES PROPERLY US PC IN DESTINATION
                                        ;DM=0,PC
003370  005067  174402                  CLR     PSW             ;KERNEL MODE!!!,PREV KERNEL MODE!!
003374  012706  001060                  MOV     #KPTR,KSP       ;SET KERNEL STACK PTR
003400  005066  177776                  CLR     -2(KSP)
003404  005237  177572                  INC     @#SR0           ;ENABLE MEMORY MANAGEMENT
003410  000277                          SCC
003412  006507          KKF10:  MFPI    PC              ;-(KSP)←PC
003414  016702  174356                  MOV     PSW,R2          ;SAVE CC'S
003420  005037  177572                  CLR     @#SR0           ;DISABLE MEMORY MANAGEMENT
003424  122702  000001                  CMPB    #C,R2           ;CHECK CC'S
003430  001401                          BEQ     .+4
003432  000000                          HLT
003434  022706  001056                  CMP     #KPTR-2,KSP     ;CHECK STACK PTR
003440  001401                          BEQ     .+4
003442  000000                          HLT                     ;ERROR! STACK NOT PUSHED
003444  022716  003414                  CMP     #KKF10+2,(KSP)  ;CHECK THAT PS WAS PUSHED ON THE STACK
003450  001401                          BEQ     .+4
003452  000000                          HLT                     ;ERROR! PC NOT PUSHED ON THE STACK
003454  104000                          SCOPE
```

```
                                        ;DM=3,PC
003456  012767  000000  174312          MOV     #0,PSW          ;KERNEL MODE!!!,PREV KERNEL MODE!!
003464  005066  177776                  CLR     -2(KSP)
003470  012737  177777  016600          MOV     #-1,@#PHYS
003476  005237  177572                  INC     @#SR0           ;ENABLE MEMORY MANAGEMENT
003502  006537  016600          KKF11:  MFPI    @#VIRT          ;-(KSP)←VIRT
003506  005037  177572                  CLR     @#SR0           ;DISABLE MEMORY MANAGEMENT
003512  005216                          INC     (KSP)           ;CHECK RESULT
003514  001401                          BEQ     .+4
003516  000000                          HLT                     ;ERROR! INCORRECT RESULT
003520  104000                          SCOPE

                                        ;DM=6,PC
003522  012767  000340  174246          MOV     #PRTY7,PSW      ;KERNEL MODE!!!,PREV KERNEL MODE!!
003530  012766  177777  177776          MOV     #-1,-2(KSP)
003536  005037  016600                  CLR     @#PHYS          ;PRESET PHYSICAL ADDRESS DATA
003542  005237  177572                  INC     @#SR0           ;ENABLE MEMORY MANAGEMENT
003546  006567  013026          KKF12:  MFPI    VIRT            ;-(KSP)←VIRT
003552  005037  177572                  CLR     @#SR0           ;DISABLE MEMORY MANAGEMENT
003556  005716                          TST     (KSP)           ;CHECK RESULT
003560  001401                          BEQ     .+4
003562  000000                          HLT                     ;ERROR! INCORRECT RESULT
003564  104000                          SCOPE

                                        ;DM=7,PC
003566  005067  174204                  CLR     PSW             ;KERNEL MODE!!!,PREV KERNEL MODE!!
003572  005066  177776                  CLR     -2(KSP)
003576  012737  016604  001004          MOV     #VIRT+4,@#TEMP  ;LOAD ADDRESS
003604  012737  177777  016604          MOV     #-1,@#PHYS+4    ;PRESET DATA
003612  005237  177572                  INC     @#SR0           ;ENABLE MEMORY MANAGEMENT
003616  000277                          SCC
003620  006577  175160          KKF13:  MFPI    @TEMP           ;-(KSP)←VIRT+4
003624  016702  174146                  MOV     PSW,R2          ;SAVE CC'S
003630  005037  177572                  CLR     @#SR0           ;DISABLE MEMORY MANAGEMENT
003634  122702  000011                  CMPB    #N+C,R2         ;CHECK CC'S
003640  001401                          BEQ     .+4
003642  000000                          HLT                     ;ERROR! INCORRECT CC'S
003644  005216                          INC     (KSP)           ;CHECK RESULT
003646  001401                          BEQ     .+4
003650  000000                          HLT                     ;ERROR! INCORRECT RESULT ON STACK
003652  104000                          SCOPE

                                        ;DM=1,PC
003654  012766  177777  177776          MOV     #-1,-2(KSP)
003662  005237  177572                  INC     @#SR0           ;ENABLE MEMORY MANAGEMENT
003666  006517                  KKF14:  MFPI    (PC)            ;PUSH NEXT WORD ON THE STACK
003670  000400                  KKF14A: BR      .+2             ;THIS DATA GOES ONTO THE STACK
003672  005037  177572                  CLR     @#SR0           ;DISABLE MEMORY MANAGEMENT
003676  023716  003670                  CMP     @#KKF14A,(KSP)  ;CHECK DATA ON THE STACK
003702  001401                          BEQ     .+4
003704  000000                          HLT                     ;ERROR! INCORRECT DATA ON STACK
003706  104000                          SCOPE
```

```
                                ;TEST THAT KERNEL CAN LOAD USER ADDRESS (VIRT)
                                ;DM=1

                                ;*********************************
          120000                        VIRT=120000     ;USER VIRTUAL ADDRESS FOR THESE TESTS
          017200                        PHYS=17200      ;CORRESPONDING PHYSICAL ADDRESS
                                ;*********************************

003710  012737  030000  177776            MOV     #KM+PUM,@#PSW   ;KERNEL MODE!!!,PREV USER MODE!!
003716  005016                            CLR     (KSP)           ;PUT 0 ON KERNEL STACK
003720  012702  120000                      MOV     #VIRT,R2                ;R2=VIRT ADDRESS
003724  012737  177777  017200            MOV     #-1,@#PHYS      ;PRESET DATA
003732  005237  177572                      INC     @#SR0           ;ENABLE KT11

003736  000277                            SCC
003740  006612                    KU1:    MTPI    (R2)            ;VIRT=(KSP)+
003742  016703  174030                    MOV     PSW,R3          ;SAVE STATUS RESULT
003746  005037  177572                    CLR     @#SR0           ;DISABLE KT11
003752  022706  001062                    CMP     #KPTR+2,KSP     ;CHECK THAT STACK POPPED
003756  001401                            BEQ     .+4
003760  000000                            HLT                     ;ERROR! INCORRECT STACK PTR
003762  122703  000005                    CMPB    #Z+C,R3         ;CHECK STATUS RESULT
003766  001401                            BEQ     .+4
003770  000000                            HLT                     ;ERROR! INCORRECT STATUS RESULT
003772  005737  017200                    TST     @#PHYS          ;CHECK RESULT
003776  001401                            BEQ     .+4
004000  000000                            HLT                     ;ERROR! INCORRECT RESULT
004002  104000                            SCOPE

                                ;DM=2
004004  012737  030000  177776            MOV     #PUM,@#PSW      ;KERNEL MODE!!!,PREV USER MODE!!
004012  012716  177777                    MOV     #-1,(KSP)       ;PUT #-1 ON KERNEL STACK
004016  012702  120000                    MOV     #VIRT,R2        ;R2=VIRT ADDRESS
004022  005037  017200                    CLR     @#PHYS          ;PRESET DATA
004026  005237  177572                    INC     @#SR0           ;ENABLE KT11

004032  006622                    KU2:    MTPI    (R2)+           ;VIRT=(KSP)+
004034  005037  177572                    CLR     @#SR0           ;DISABLE KT11
004040  005237  017200                    INC     @#PHYS          ;CHECK RESULT
004044  001401                            BEQ     .+4
004046  000000                            HLT                     ;ERROR! INCORRECT RESULT
004050  022702  120002                    CMP     #VIRT+2,R2      ;CHECK AUTO-INCREMENT
004054  001401                            BEQ     .+4
004056  000000                            HLT                     ;ERROR! AUTO-INCREMENT FAILED
004060  005067  173712                    CLR     PSW
004064  104000                            SCOPE
                                ;DM=3
004066  012737  030340  177776            MOV     #PUM+PRTY7,@#PSW        ;KERNEL MODE!!!,PREV USER MODE!!
004074  012716  177777                    MOV     #-1,(KSP)       ;PUT #-1 ON KERNEL STACK
004100  012702  001004                    MOV     #TEMP,R2        ;LOAD INDIRECT ADDRESS
004104  012712  120004                    MOV     #VIRT+4,(R2)    ;LOAD ADDRESS
004110  005037  017204                    CLR     @#PHYS+4        ;PRESET DATA
004114  005237  177572                    INC     @#SR0           ;ENABLE KT11
```

```
004120  006632                    KU3:    MTPI    @(R2)+          ;VIRT+4-(KSP)+
004122  005037  177572                    CLR     @#SR0           ;DISABLE KT11
004126  005237  017204                    INC     @#PHYS+4                ;CHECK RESULT
004132  001401                            BEQ     .+4
004134  000000                            HLT                     ;ERROR! INCORRECT RESULT
004136  104000                            SCOPE

                                ;DM=4
004140  012737  030000  177776            MOV     #KM+PUM,@#PSW   ;KERNEL MODE!!!,PREV USER MODE!!
004146  005016                            CLR     (KSP)           ;PUT 0 ON KERNEL STACK
004150  012704  120002                    MOV     #VIRT+2,R4      ;LOAD ADDRESS
004154  012737  177777  017200            MOV     #-1,@#PHYS      ;PRESET DATA
004162  005237  177572                    INC     @#SR0           ;ENABLE KT11

004166  006644                    KU4:    MTPI    -(R4)           ;VIRT+(KSP)+
004170  005037  177572                    CLR     @#SR0           ;DISABLE KT11
004174  022704  120000                    CMP     #VIRT,R4                ;CHECK AUTO-DECREMENT
004200  001401                            BEQ     .+4
004202  000000                            HLT                     ;ERROR! AUTO-DECREMENT FAILED
004204  005737  017200                    TST     @#PHYS          ;CHECK RESULT
004210  001401                            BEQ     .+4
004212  000000                            HLT                     ;ERROR! INCORRECT RESULT
004214  104000                            SCOPE

                                ;DM=5
004216  012737  030000  177776            MOV     #PUM,@#PSW      ;KERNEL MODE!!!,PREV USER MODE!!
004224  012716  177777                    MOV     #-1,(KSP)       ;PUT #-1 ON KERNEL STACK
004230  012702  001010                    MOV     #TEMP+4,R2      ;LOAD INDIRECT ADDRESS
004234  012767  120000  174544            MOV     #VIRT,TEMP+2    ;LOAD ADDRESS
004242  005037  017200                    CLR     @#PHYS          ;PRESET DATA
004246  005237  177572                    INC     @#SR0           ;ENABLE KT11

004252  006652                    KU5:    MTPI    @-(R2)          ;VIRT+(KSP)+
004254  005037  177572                    CLR     @#SR0           ;DISABLE KT11
004260  005237  017200                    INC     @#PHYS          ;CHECK RESULT
004264  001401                            BEQ     .+4
004266  000000                            HLT                     ;ERROR! INCORRECT RESULT
004270  005067  173502                    CLR     PSW
004274  104000                            SCOPE

                                ;DM=6
004276  012737  030000  177776            MOV     #PUM,@#PSW      ;KERNEL MODE!!!,PREV USER MODE!!
004304  005016                            CLR     (KSP)           ;PUT 0 ON KERNEL STACK
004306  012702  000002                    MOV     #2,R2           ;LOAD INDEX REGISTER
004312  012767  177777  012662            MOV     #-1,PHYS+2      ;PRESET DATA
004320  005237  177572                    INC     @#SR0           ;ENABLE KT11

004324  006662  120000            KU6:    MTPI    VIRT(R2)        ;VIRT+2-(KSP)+
004330  016700  173442                    MOV     PSW,R0          ;SAVE STATUS RESULT
004334  005037  177572                    CLR     @#SR0           ;DISABLE KT11
004340  022706  001062                    CMP     #KPTR+2,KSP     ;CHECK THAT STACK POINTER POPPED
004344  001401                            BEQ     .+4
004346  000000                            HLT                     ;ERROR! INCORRECT STACK PTR
```

```
004350  122700  000004          CMPB    #Z,R0    ;CHECK STATUS RESULT
004354  001401                  BEQ     .+4
004356  000000                  HLT                      ;ERROR: INCORRECT STATUS RESULT
004360  005737  017202          TST     @#PHYS+2         ;CHECK RESULT
004364  001401                  BEQ     .+4
004366  000000                  HLT                      ;ERROR: INCORRECT RESULT
004370  104000                  SCOPE

                        ;DM=7
004372  012737  030000  177776  MOV     #KM+PUM,@#PSW    ;KERNEL MODE!!!,PREV USER MODE!!
004400  012716  177777          MOV     #-1,(KSP)        ;PUT #-1 ON KERNEL STACK
004404  012702  000002          MOV     #2,R2            ;LOAD INDEX REGISTER
004410  012737  120000  001010  MOV     #VIRT,@#TEMP+4   ;LOAD ADDRESS
004416  005037  017200          CLR     @#PHYS           ;PRESET DATA
004422  005237  177572          INC     @#SR0            ;ENABLE KT11

004426  006672  001006  KU7:    MTPI    @TEMP+2(R2)      ;VIRT=(KSP)+
004432  005037  177572          CLR     @#SR0            ;DISABLE KT11
004436  005237  017200          INC     @#PHYS           ;CHECK RESULT
004442  001401                  BEQ     .+4
004444  000000                  HLT                      ;ERROR: INCORRECT RESULT
004446  104000                  SCOPE

                        ;TEST THAT MTPI CAN LOAD USER
                        ;DM=0,PC
004450  012737  030000  177776          MOV     #KM+PUM,@#PSW    ;KERNEL MODE!!!,PREV USER MODE!!
004456  012716  004472                  MOV     #KU10A,(KSP)     ;PUT NEW PC ON STACK AS DATA
004462  005237  177572                  INC     @#SR0            ;ENABLE KT11

004466  006607          KU10:   MTPI    PC               ;PC=(KSP)+
004470  000000                  HLT                      ;ERROR: MTPI DID NOT LOAD NEW PC
004472  005037  177572  KU10A:  CLR     @#SR0            ;DISABLE KT11
004476  104000                  SCOPE

                        ;DM=2,PC
004500  012737  030000  177776          MOV     #KM+PUM,@#PSW    ;KERNEL MODE!!!,PREV USER MODE!!
004506  012716  177777                  MOV     #-1,(KSP)        ;PUT #-1 ON KERNEL STACK
004512  012767  004532  173530          MOV     #KU11A,MMVEC     ;LOAD SEG ERR VECTOR
004520  005237  177572                  INC     @#SR0            ;ENABLE KT11

004524  006627          KU11:   MTPI    (PC)+            ;((PC)+=(KSP)+, SHOULD ABORT
004526  000000                  HLT                      ;ERROR: DID NOT ABORT AND PC DID NOT
                                                         ;AUTO-INCREMENT
004530  000000                  HLT                      ;ERROR: DID NOT ABORT
004532  005037  177572  KU11A:  CLR     @#SR0            ;DISABLE KT11
004536  022706  001056          CMP     #KPTR-2,KSP      ;CHECK THAT STACK PTR WAS PUSHED TWICE
004542  001401                  BEQ     .+4
004544  000000                  HLT                      ;ERROR: INCORRECT STACK PTR ON ERROR ABORT
004546  012767  000462  173474  MOV     #MMERR,MMVEC
004554  104000                  SCOPE

                        ;DM=3,PC
004556  012737  030000  177776          MOV     #PUM,@#PSW       ;KERNEL MODE!!!,PREV USER MODE!!
004564  012716  177777                  MOV     #-1,(KSP)        ;PUT #-1 ON KERNEL STACK
```

```
004570  005037  017200                CLR     @#PHYS
004574  005237  177572                INC     @#SR0                   ;ENABLE KT11

004600  006637  120000        KU12:   MTPI    @#VIRT                  ;VIRT←(KSP)+
004604  016700  173166                MOV     PSW,R0                  ;SAVE STATUS RESULT
004610  005037  177572                CLR     @#SR0                   ;DISABLE KT11
004614  122700  000010                CMPB    #N,R0   ;CHECK STATUS RESULT
004620  001401                        BEQ     .+4
004622  000000                        HLT                             ;ERROR!  INCORRECT STATUS RESULT
004624  005267  012350                INC     PHYS                    ;CHECK RESULT
004630  001401                        BEQ     .+4
004632  000000                        HLT                             ;ERROR!  INCORRECT RESULT
004634  005067  173136                CLR     PSW
004640  104000                        SCOPE

                              ;DM=4,PC
004642  012737  030000  177776        MOV     #KM+PUM,@#PSW   ;KERNEL MODE!!!,PREV USER MODE!!
004650  005016                        CLR     (KSP)                   ;PUT 0 ON KERNEL STACK
004652  016702  000012                MOV     KU13,R2 ;SAVE MTPI INSTRUCTION
004656  012767  004674  173364        MOV     #KU13A,MMVEC    ;LOAD SEG ERR VECTOR
004664  005237  177572                INC     @#SR0                   ;ENABLE KT11

004670  006647                KU13:   MTPI    -(PC)                   ;-(PC)←(KSP)+
004672  000000                        HLT                             ;ERROR! FAILED TO ABORT
004674  005037  177572        KU13A:  CLR     @#SR0                   ;DISABLE KT11
004700  010267  177764                MOV     R2,KU13 ;RESTORE INSTRUCTION
004704  012767  000462  173336        MOV     #MMERR,MMVEC
004712  104000                        SCOPE

                              ;DM=6,PC
004714  012737  030340  177776        MOV     #PUM+PRTY7,@#PSW        ;KERNEL MODE!!!,PREV USER MODE!!
004722  005016                        CLR     (KSP)                   ;PUT 0 ON KERNEL STACK
004724  012767  177777  012252        MOV     #-1,PHYS+4
004732  005237  177572                INC     @#SR0                   ;ENABLE KT11

004736  000277                        SCC
004740  006667  113040        KU14:   MTPI    VIRT+4                  ;VIRT+4←(KSP)+
004744  016703  173026                MOV     PSW,R3                  ;SAVE STATUS RESULT
004750  005037  177572                CLR     @#SR0                   ;DISABLE KT11
004754  022706  001062                CMP     #KPTR+2,KSP             ;CHECK THAT STACK PTR POPPED
004760  001401                        BEQ     .+4
004762  000000                        HLT                             ;ERROR!  INCORRECT STACK PTR
004764  122703  000345                CMPB    #PRTY7+Z+C,R3   ;CHECK STATUS RESULT
004770  001401                        BEQ     .+4
004772  000000                        HLT                             ;ERROR!  INCORRECT STATUS RESULT
004774  005737  017204                TST     @#PHYS+4                ;CHECK RESULT
005000  001401                        BEQ     .+4
005002  000000                        HLT                             ;ERROR!  INCORRECT RESULT
005004  104000                        SCOPE

                              ;DM=7,PC

005006  012737  030000  177776        MOV     #KM+PUM,@#PSW   ;KERNEL MODE!!!,PREV USER MODE!!
005014  012716  177777                MOV     #-1,(KSP)               ;PUT #-1 ON KERNEL STACK
```

```
005020  012737  120004  001004          MOV     #VIRT+4,@#TEMP  ;LOAD ADDRESS
005026  005037  017204                  CLR     @#PHYS+4                ;PRESET DATA
005032  005237  177572                  INC     @#SR0           ;ENABLE KT11

005036  006677  173742          KU15:   MTPI    @TEMP           ;VIRT+4-(KSP)+
005042  005037  177572                  CLR     @#SR0           ;DISABLE KT11
005046  005237  017204                  INC     @#PHYS+4                ;CHECK RESULT
005052  001401                          BEQ     .+4
005054  000000                          HLT                     ;ERROR INCORRECT RESULT
005056  104000                          SCOPE


                                ;CHECK THAT MTPI CAN SET USER STACK PTR & PUSH DATA ONTO USER STACK
005060  012737  030000  177776          MOV     #PUM,@#PSW      ;KERNEL MODE!!!,PREV USER MODE!!
005066  012746  120000                  MOV     #VIRT,-(KSP)
005072  005046                          CLR     -(KSP)          ;PUT DATA ON YHE STACK
005074  012746  120000                  MOV     #VIRT,-(KSP)
005100  012737  177777  017200          MOV     #-1,@#PHYS      ;PRESET STACK DATA
005106  005237  177572                  INC     @#SR0           ;ENABLE KT11

005112  006606                  KU16:   MTPI    USP             ;USP-(KSP)+
005114  006636                          MTPI    @(KSP)+         ;VIRT-(KSP)+
005116  005037  177572                  CLR     @#SR0           ;DISABLE KT11
005122  106506                          MFPD    USP             ;GET USER STACK PTR
005124  022716  120000                  CMP     #VIRT,(KSP)     ;CHECK THAT MTPI  USP SET USER STACK PTR
005130  001401                          BEQ     .+4
005132  000000                          HLT                     ;ERROR! MTPI  USP FAILED
005134  005737  017200                  TST     @#PHYS          ;CHECK THAT MTPI  @(KSP)+ PUT THE
005140  001401                          BEQ     .+4             ;CORRECT DATA ONTO THE USER STACK
005142  000000                          HLT                     ;ERROR! MTPI  @(KSP)+ FAILED
005144  022706  001056                  CMP     #KPTR-2,KSP     ;CHECK KERNEL STACK PTR AFTER TEST
005150  001401                          BEQ     .+4
005152  000000                          HLT                     ;ERROR! INCORRECT KERNEL STACK PTR
005154  104000                          SCOPE
                                ;TEST MFPI INSTRUCTION KERNEL MODE PREVIOUS USER MODE
                                ;TEST THAT MFPI CAN GET DATA FROM A GENERAL REGISTER (R1).
005156  012767  030000  172612          MOV     #PUM,PSW        ;KERNEL MODE!!!,PREV USER MODE!!
005164  012766  177777  177776          MOV     #-1,-2(KSP)
005172  005000                          CLR     R0              ;PRESET REGISTER
005174  005237  177572                  INC     @#SR0           ;ENABLE MEMORY MANAGEMENT
005200  000277                          SCC                     ;PRESET CC'S
005202  006500                  KUF0:   MFPI    R0              ;-(KSP)-(R1)
005204  016704  172566                  MOV     PSW,R4          ;SAVE CC'S
005210  005037  177572                  CLR     @#SR0           ;DISABLE MEMORY MANAGEMENT
005214  122704  000005                  CMPB    #Z+C,R4 ;CHECK CC'S
005220  001401                          BEQ     .+4
005222  000000                          HLT                     ;ERROR! INCORRECT CC'S
005224  022706  001056                  CMP     #KPTR-2,KSP     ;CHECK THAT STACK PTR WAS PUSHED
005230  001401                          BEQ     .+4
005232  000000                          HLT                     ;ERROR! INCORRECT STACK PTR
005234  005716                          TST     (KSP)           ;CHECK RESULT
005236  001401                          BEQ     .+4
005240  000000                          HLT                     ;ERROR! INCORRECT RESULT
005242  005067  172530                  CLR     PSW
```

```
005246  104000                          SCOPE

                                ;TEST THAT MFPI CAN GET DATA FROM A KERNEL VIRTUAL ADDRESS
                                ;DM=1
005250  012767  030000  172520          MOV     #KM+PUM,PSW     ;KERNEL MODE!!!,PREV USER MODE!!
005256  005066  177776                  CLR     -2(KSP)
005262  012702  120000                  MOV     #VIRT,R2        ;R2=VIRTUAL ADDRESS
005266  012737  177777  017200          MOV     #-1,@#PHYS      ;PRESET PHYSICAL ADDRESS
005274  005237  177572                  INC     @#SR0           ;ENABLE MEMORY MANAGEMENT
005300  000277                          SCC                     ;PRESET CC'S
005302  006512                  KUF1:   MFPI    (R2)            ;-(KSP)<-(R2)
005304  016703  172466                  MOV     PSW,R3          ;SAVE CC'S
005310  005037  177572                  CLR     @#SR0           ;DISABLE MEMORY MANAGEMENT
005314  122703  000011                  CMPB    #N+C,R3         ;CHECK CC'S
005320  001401                          BEQ     .+4
005322  000000                          HLT                     ;ERROR! INCORRECT CC'S
005324  022706  001056                  CMP     #KPTR-2,KSP     ;CHECK THAT STACK WAS PUSHED
005330  001401                          BEQ     .+4
005332  000000                          HLT                     ;ERROR! INCORRECT STACK PTR
005334  005216                          INC     (KSP)           ;CHECK RESULT
005336  001401                          BEQ     .+4
005340  000000                          HLT                     ;ERROR! INCORRECT RESULT
005342  104000                          SCOPE

                                ;DM=2
005344  012767  030000  172424          MOV     #PUM,PSW        ;KERNEL MODE!!!,PREV USER MODE!!
005352  012766  177777  177776          MOV     #-1,-2(KSP)
005360  012702  120000                  MOV     #VIRT,R2        ;R2=VIRTUAL ADDRESS
005364  005037  017200                  CLR     @#PHYS          ;PRESET PHYSICAL ADDRESS
005370  005237  177572                  INC     @#SR0           ;ENABLE MEMORY MANAGEMENT
005374  006522                  KUF2:   MFPI    (R2)+           ;-(KSP)<-VIRT
005376  005037  177572                  CLR     @#SR0           ;DISABLE MEMORY MANAGEMENT
005402  005716                          TST     (KSP)           ;CHECK RESULT
005404  001401                          BEQ     .+4
005406  000000                          HLT                     ;ERROR! INCORRECT RESULT ON STACK
005410  022702  120002                  CMP     #VIRT+2,R2      ;CHECK AUTO INCREMENT
005414  001401                          BEQ     .+4
005416  000000                          HLT                     ;ERROR! AUTO INCREMENT FAILED
005420  005067  172352                  CLR     PSW
005424  104000                          SCOPE

                                ;DM=3
005426  012767  030000  172342          MOV     #KM+PUM,PSW     ;KERNEL MODE!!!,PREV USER MODE!!
005434  005066  177776                  CLR     -2(KSP)
005440  012702  001004                  MOV     #TEMP,R2        ;LOAD INDIRECT ADDRESS
005444  012737  120002  001004          MOV     #VIRT+2,@#TEMP  ;LOAD ADDRESS
005452  012737  177777  017202          MOV     #-1,@#PHYS+2    ;PRESET DATA
005460  005237  177572                  INC     @#SR0           ;ENABLE MEMORY MANAGEMENT
005464  006532                  KUF3:   MFPI    @(R2)+          ;-(KSP)<-VIRT+2
005466  005037  177572                  CLR     @#SR0           ;DISABLE MEMORY MANAGEMENT
005472  005216                          INC     (KSP)           ;CHECK RESULT
005474  001401                          BEQ     .+4
005476  000000                          HLT                     ;ERROR! INCORRECT RESULT
005500  104000                          SCOPE
```

```
                                        ;DM=4
005502  012767  030000  172266           MOV     #KM+PUM,PSW     ;KERNEL MODE!!!,PREV USER MODE!!
005510  012766  177777  177776           MOV     #-1,-2(KSP)
005516  012704  120002                   MOV     #VIRT+2,R4      ;R4=VIRTUAL ADDRESS+2
005522  005037  017200                   CLR     @#PHYS          ;PRESET PHYSICAL ADDRESS DATA
005526  005237  177572                   INC     @#SR0           ;ENABLE MEMORY MANAGEMENT
005532  006544                   KUF4:   MFPI    -(R4)           ;=(KSP)<-VIRT
005534  005037  177572                   CLR     @#SR0           ;DISABLE MEMORY MANAGEMENT
005540  022704  120000                   CMP     #VIRT,R4        ;CHECK AUTO-DECREMENT
005544  001401                           BEQ     .+4
005546  000000                           HLT                     ;ERROR! AUTO-DECREMENT FAILED
005550  005716                           TST     (KSP)           ;CHECK RESULT
005552  001401                           BEQ     .+4
005554  000000                           HLT                     ;ERROR! INCORRECT RESULT
005556  104000                           SCOPE

                                        ;DM=5
005560  012767  030000  172210           MOV     #PUM,PSW        ;KERNEL MODE!!!,PREV USER MODE!!
005566  005066  177776                   CLR     -2(KSP)
005572  012701  001006                   MOV     #TEMP+2,R1      ;R1=INDIRECT ADDRESS
005576  012737  120004  001004           MOV     #VIRT+4,@#TEMP  ;LOAD ADDRESS
005604  012737  177777  017204           MOV     #-1,@#PHYS+4    ;PRESET PHYSICAL ADDRESS DATA
005612  005237  177572                   INC     @#SR0           ;ENABLE MEMORY MANAGEMENT
005616  006551                   KUF5:   MFPI    @-(R1)          ;=(KSP)<-VIRT+4
005620  005037  177572                   CLR     @#SR0           ;DISABLE MEMORY MANAGEMENT
005624  005216                           INC     (KSP)           ;CHECK RESULT
005626  001401                           BEQ     .+4
005630  000000                           HLT
005632  005067  172140                   CLR     PSW
005636  104000                           SCOPE

                                        ;DM=6
005640  012767  030000  172130           MOV     #PUM,PSW        ;KERNEL MODE!!!,PREV USER MODE!!
005646  012766  177777  177776           MOV     #-1,-2(KSP)
005654  012702  000002                   MOV     #2,R2           ;LOAD INDEX REGISTER
005660  005037  017202                   CLR     @#PHYS+2        ;PRESET PHYSICAL ADDRESS DATA
005664  005237  177572                   INC     @#SR0           ;ENABLE MEMORY MANAGEMENT
005670  006562  120000           KUF6:   MFPI    VIRT(R2)        ;=(KSP)<-VIRT-2
005674  005037  177572                   CLR     @#SR0           ;DISABLE MEMORY MANAGEMENT
005700  022706  001056                   CMP     #KPTR-2,KSP     ;CHECK STACK PTR
005704  001401                           BEQ     .+4
005706  000000                           HLT                     ;ERROR! INCORRECT STACK PTR
005710  005716                           TST     (KSP)           ;CHECK RESULT
005712  001401                           BEQ     .+4
005714  000000                           HLT                     ;ERROR! INCORRECT RESULT
005716  005067  172054                   CLR     PSW
005722  104000                           SCOPE

                                        ;DM=7
005724  012767  030000  172044           MOV     #KM+PUM,PSW     ;KERNEL MODE!!!,PREV USER MODE!!
005732  005066  177776                   CLR     -2(KSP)
005736  012702  177774                   MOV     #-4,R2          ;LOAD INDEX REGISTER
005742  012737  120000  001004           MOV     #VIRT,@#TEMP    ;LOAD ADDRESS
```

```
005750  012737  177777  017200          MOV     #-1,@#PHYS      ;CLEAR PHYSICAL ADDRESS DATA
005756  005237  177572                  INC     @#SR0           ;ENABLE MEMORY MANAGEMENT
005762  006572  001010          KUF7:   MFPI    @TEMP+4(R2)     ;-(KSP)←VIRT
005766  005037  177572                  CLR     @#SR0           ;DISABLE MEMORY MANAGEMENT
005772  005216                  INC     (KSP)           ;CHECK RESULT
005774  001401                  BEQ     .+4
005776  000000                  HLT                             ;ERROR! INCORRECT RESULT
006000  104000                  SCOPE

                        ;TEST THAT MFPI OPERATES PROPERLY US PC IN DESTINATION
                        ;DM=0,PC
006002  012767  030000  171766          MOV     #KM+PUM,PSW     ;KERNEL MODE!!!,PREV USER MODE!!
006010  005066  177776                  CLR     -2(KSP)
006014  005237  177572                  INC     @#SR0           ;ENABLE MEMORY MANAGEMENT
006020  000277                  SCC
006022  006507          KUF10:  MFPI    PC              ;-(KSP)←PC
006024  016702  171746                  MOV     PSW,R2          ;SAVE CC'S
006030  005037  177572                  CLR     @#SR0           ;DISABLE MEMORY MANAGEMENT
006034  122702  000001                  CMPB    #C,R2           ;CHECK CC'S
006040  001401                  BEQ     .+4
006042  000000                  HLT
006044  022706  001056                  CMP     #KPTR-2,KSP     ;CHECK STACK PTR
006050  001401                  BEQ     .+4
006052  000000                  HLT                             ;ERROR! STACK NOT PUSHED
006054  022716  006024                  CMP     #KUF10+2,(KSP)  ;CHECK THAT PC WAS PUSHED ON THE STACK
006060  001401                  BEQ     .+4
006062  000000                  HLT                             ;ERROR! PC NOT PUSHED ON THE STACK
006064  104000                  SCOPE



                        ;DM=3,PC
006066  012767  030000  171702          MOV     #PUM,PSW        ;KERNEL MODE!!!,PREV USER MODE!!
006074  005066  177776                  CLR     -2(KSP)
006100  012737  177777  017200          MOV     #-1,@#PHYS
006106  005237  177572                  INC     @#SR0           ;ENABLE MEMORY MANAGEMENT
006112  006537  120000          KUF11:  MFPI    @#VIRT          ;-(KSP)←VIRT
006116  005037  177572                  CLR     @#SR0           ;DISABLE MEMORY MANAGEMENT
006122  005216                  INC     (KSP)           ;CHECK RESULT
006124  001401                  BEQ     .+4
006126  000000                  HLT                             ;ERROR! INCORRECT RESULT
006130  104000                  SCOPE

                        ;DM=6,PC
006132  012767  030340  171636          MOV     #PUM+PRTY7,PSW  ;KERNEL MODE!!!, PREV USER MODE!!
006140  012766  177777  177776          MOV     #-1,-2(KSP)
006146  005037  017200                  CLR     @#PHYS          ;PRESET PHYSICAL ADDRESS DATA
006152  005237  177572                  INC     @#SR0           ;ENABLE MEMORY MANAGEMENT
006156  006567  111616          KUF12:  MFPI    VIRT            ;-(KSP)←VIRT
006162  005037  177572                  CLR     @#SR0           ;DISABLE MEMORY MANAGEMENT
006166  005716                  TST     (KSP)           ;CHECK RESULT
006170  001401                  BEQ     .+4
006172  000000                  HLT                             ;ERROR! INCORRECT RESULT
006174  104000                  SCOPE
```

```
                                        ;DM=7,PC
006176  012767  030000  171572           MOV     #KM+PUM,PSW     ;KERNEL MODE!!!,PREV USER MODE!!
006204  005066  177776                   CLR     -2(KSP)
006210  012737  120004  001004           MOV     #VIRT+4,@#TEMP  ;LOAD ADDRESS
006216  012737  177777  017204           MOV     #-1,@#PHYS+4    ;PRESET DATA
006224  005237  177572                   INC     @#SR0           ;ENABLE MEMORY MANAGEMENT
006230  000277                           SCC
006232  006577  172546           KUF13:  MFPI    @TEMP           ;-(KSP)<-VIRT+4
006236  016702  171534                   MOV     PSW,R2          ;SAVE CC'S
006242  005037  177572                   CLR     @#SR0           ;DISABLE MEMORY MANAGEMENT
006246  122702  000011                   CMPB    #N+C,R2         ;CHECK CC'S
006252  001401                           BEQ     .+4
006254  000000                           HLT                     ;ERROR! INCORRECT CC'S
006256  005216                           INC     (KSP)           ;CHECK RESULT
006260  001401                           BEQ     .+4
006262  000000                           HLT                     ;ERROR! INCORRECT RESULT ON STACK
006264  104000                           SCOPE

006266  012767  030000  171502           MOV     #KM+PUM,PSW     ;KERNEL MODE!!!,PREV USER MODE!!
006274  012716  120000                   MOV     #VIRT,(KSP)
006300  005037  017200                   CLR     @#PHYS
006304  005237  177572                   INC     @#SR0           ;ENABLE MEMORY MANAGEMENT
006310  006576  000000           KUF14:  MFPI    @(KSP)          ;-(KSP)<-VIRT
006314  005037  177572                   CLR     @#SR0           ;DISABLE MEMORY MANAGEMENT
006320  005737  001056                   TST     @#KPTR-2        ;CHECK DATA ON THE STACK
006324  001401                           BEQ     .+4
006326  000000                           HLT                     ;ERROR! INCORRECT DATA ON THE STACK
006330  104000                           SCOPE

                                        ;BEGIN TESTING IN USER MODE
                                        ;NOTE: ALL HLT (HALT) INSTRUCTIONS WILL TRAP TO LOC 10. THE PROGRAM WILL
                                        ;ALLOW THE TRAP,ADJUST THE PC AND RETURN TO THE HLT IN KERNEL MODE. THE
                                        ;USER STACK POINTER IS NOT AFFECTED BY THIS TRAP. THE USER STACK POINTER
                                        ;IS AT PHYSICAL 0600.

006332  012706  000600                   MOV     #UPTR,USP       ;SET USER STACK PTR
006336  000240           USRTST: NOP                             ;BEGIN TESTS IN USER MODE
006340  012767  077406  171232           MOV     #77406,UPDR0    ;RW,UP 200 BLOCKS
006346  012767  077406  171242           MOV     #77406,UPDR7    ;RW, UP 200 BLOCKS
006354  012767  007600  171274           MOV     #7600,UPAR7

                                        ;TESTS UU0-UU6 TEST THE MTPI INSTRUCTION IN USER MODE, PREV USER MODE.
                                        ;TEST THAT MTPI CAN LOAD A GENERAL REGISTER (R2)
006362  012737  170340  177776           MOV     #UM+PUM+PRTY7,@#PSW     ;USER MODE!!!,PREV USER MODE!!
006370  005016                           CLR     (USP)
006372  012702  177777                   MOV     #-1,R2          ;PRESET REGISTER
006376  005237  177572                   INC     @#SR0           ;ENABLE KT11

006402  006602           UU0:    MTPI    R2                      ;R2<-(USP)+
006404  016703  171366                   MOV     PSW,R3          ;SAVE STATUS RESULT
006410  005037  177572                   CLR     @#SR0           ;DISABLE KT11
006414  022706  000602                   CMP     #UPTR+2,USP     ;CHECK THAT STACK POPPED
006420  001401                           BEQ     .+4
```

```
006422  000000                          HLT                     ;ERROR!  INCORRECT STACK PTR
006424  122703  000344                  CMPB    #PRTY7+2,R3     ;CHECK STATUS RESULT
006430  001401                          BEQ     .+4
006432  000000                          HLT                     ;ERROR! INCORRECT STATUS RESULT
006434  005702                          TST     R2              ;CHECK RESULT
006436  001401                          BEQ     .+4
006440  000000                          HLT                     ;ERROR!  INCORRECT RESULT
006442  104000                          SCOPE


                                ;TEST THAT USER CAN LOAD USER ADDRESS (VIRT)

                                ;*********************************
        120000                          VIRT=120000     ;USER VIRTUAL ADDRESS FOR THESE TESTS
        017200                          PHYS=17200      ;CORRESPONDING PHYSICAL ADDRESS
                                ;*********************************

                                ;DM=2
006444  012737  170000  177776          MOV     #UM+PUM,@#PSW   ;USER MODE!!!,PREV USER MODE!!
006452  012716  177777                  MOV     #-1,(USP)
006456  012702  120000                  MOV     #VIRT,R2        ;R2=VIRT ADDRESS
006462  005037  017200                  CLR     @#PHYS          ;PRESET DATA
006466  005237  177572                  INC     @#SR0           ;ENABLE KT11

006472  006622                  UU1:    MTPI    (R2)+           ;VIRT=(USP)+
006474  005037  177572                  CLR     @#SR0           ;DISABLE KT11
006500  005237  017200                  INC     @#PHYS          ;CHECK RESULT
006504  001401                          BEQ     .+4
006506  000000                          HLT                     ;ERROR! INCORRECT RESULT
006510  022702  120002                  CMP     #VIRT+2,R2      ;CHECK AUTO-INCREMENT
006514  001401                          BEQ     .+4
006516  000000                          HLT                     ;ERROR! AUTO-INCREMENT FAILED
006520  005067  171252                  CLR     PSW
006524  104000                          SCOPE
                                ;DM=4
006526  012737  170000  177776          MOV     #UM+PUM,@#PSW   ;USER MODE!!!,PREV USER MODE!!
006534  005016                          CLR     (USP)
006536  012704  120002                  MOV     #VIRT+2,R4      ;LOAD ADDRESS
006542  012737  177777  017200          MOV     #-1,@#PHYS      ;PRESET DATA
006550  005237  177572                  INC     @#SR0           ;ENABLE KT11

006554  006644                  UU2:    MTPI    -(R4)           ;VIRT=(USP)+
006556  005037  177572                  CLR     @#SR0           ;DISABLE KT11
006562  022704  120000                  CMP     #VIRT,R4                ;CHECK AUTO-DECREMENT
006566  001401                          BEQ     .+4
006570  000000                          HLT                     ;ERROR! AUTO-DECREMENT FAILED
006572  005737  017200                  TST     @#PHYS          ;CHECK RESULT
006576  001401                          BEQ     .+4
006600  000000                          HLT                     ;ERROR! INCORRECT RESULT
006602  104000                          SCOPE

                                ;DM=6
006604  012737  170000  177776          MOV     #UM+PUM,@#PSW   ;USER MODE!!!,PREV USER MODE!!
006612  005016                          CLR     (USP)
```

```
006614  012702  000002                MOV     #2,R2           ;LOAD INDEX REGISTER
006620  012767  177777  010354        MOV     #-1,PHYS+2      ;PRESET DATA
006626  005237  177572                INC     @#SR0           ;ENABLE KT11

006632  006662  120000        UU3:    MTPI    VIRT(R2)        ;VIRT+2+(USP)+
006636  016700  171134                MOV     PSW,R0          ;SAVE STATUS RESULT
006642  005037  177572                CLR     @#SR0           ;DISABLE KT11
006646  022706  000602                CMP     #UPTR+2,USP     ;CHECK THAT STACK POINTER POPPED
006652  001401                        BEQ     .+4
006654  000000                        HLT                     ;ERROR! INCORRECT STACK PTR
006656  122700  000004                CMPB    #Z,R0   ;CHECK STATUS RESULT
006662  001401                        BEQ     .+4
006664  000000                        HLT                     ;ERROR! INCORRECT STATUS RESULT
006666  005737  017202                TST     @#PHYS+2                ;CHECK RESULT
006672  001401                        BEQ     .+4
006674  000000                        HLT                     ;ERROR! INCORRECT RESULT
006676  104000                        SCOPE

                                ;TEST THAT MTPI CAN LOAD PC
                                ;DM=0,PC
006700  012737  170000  177776        MOV     #UM+PUM,@#PSW   ;USER MODE!!!,PREV USER MODE!!
006706  012716  006722                MOV     #UU4A,(USP)     ;PUT NEW PC ON STACK AS DATA
006712  005237  177572                INC     @#SR0           ;ENABLE KT11

006716  006607                UU4:    MTPI    PC              ;PC+(USP)+
006720  000000                        HLT                     ;ERROR! MTPI DID NOT LOAD PC
006722  005037  177572        UU4A:   CLR     @#SR0           ;DISABLE KT11
006726  104000                        SCOPE

                                ;DM=3,PC
006730  012737  170000  177776        MOV     #UM+PUM,@#PSW   ;USER MODE!!!,PREV USER MODE!!
006736  012716  177777                MOV     #-1,(USP)
006742  005037  017200                CLR     @#PHYS
006746  005237  177572                INC     @#SR0           ;ENABLE KT11

006752  006637  120000        UU5:    MTPI    @#VIRT          ;VIRT+(USP)+
006756  016700  171014                MOV     PSW,R0          ;SAVE STATUS RESULT
006762  005037  177572                CLR     @#SR0           ;DISABLE KT11
006766  122700  000010                CMPB    #N,R0   ;CHECK STATUS RESULT
006772  001401                        BEQ     .+4
006774  000000                        HLT                     ;ERROR!  INCORRECT STATUS RESULT
006776  005267  010176                INC     PHYS            ;CHECK RESULT
007002  001401                        BEQ     .+4
007004  000000                        HLT                     ;ERROR!  INCORRECT RESULT
007006  005067  170764                CLR     PSW
007012  104000                        SCOPE

                                ;DM=7,PC
007014  012737  170000  177776        MOV     #UM+PUM,@#PSW   ;USER MODE!!!,PREV USER MODE!!
007022  012716  177777                MOV     #-1,(USP)
007026  012737  120004  001004        MOV     #VIRT+4,@#TEMP  ;LOAD ADDRESS
007034  005037  017204                CLR     @#PHYS+4        ;PRESET DATA
007040  005237  177572                INC     @#SR0           ;ENABLE KT11
```

```
007044  006677  171734          UU6:    MTPI    @TEMP           ;VIRT+4=(USP)+
007050  005037  177572                  CLR     @#SR0           ;DISABLE KT11
007054  005237  017204                  INC     @#PHYS+4        ;CHECK RESULT
007060  001401                          BEQ     .+4
007062  000000                          HLT                     ;ERROR INCORRECT RESULT
007064  104000                          SCOPE
                                ;TEST MFPI INSTRUCTION USER MODE PREVIOUS USER MODE
007066  012767  170000  170702          MOV     #UM+PUM,PSW     ;USER MODE!!!,PREV USER MODE!!
007074  012703  177777                  MOV     #-1,R3          ;PRESET GENERAL REGISTER
007100  005237  177572                  INC     @#SR0           ;ENABLE MEMORY MANAGEMENT
007104  006503                  UUF0:   MFPI    R3              ;-(USP)=R3
007106  016702  170664                  MOV     PSW,R2          ;SAVE STATUS AFTER MFPI
007112  005037  177572                  CLR     @#SR0           ;DISABLE MEMORY MANAGEMENT
007116  022702  170010                  CMP     #UM+PUM+N,R2    ;CHECK STATUS AFTER MFPI
007122  001401                          BEQ     .+4
007124  000000                          HLT                     ;ERROR! INCORRECT STATUS AFTER MFPI
007126  022706  000576                  CMP     #UPTR-2,USP     ;CHECK THAT STACK WAS PUSHED
007132  001401                          BEQ     .+4
007134  000000                          HLT                     ;ERROR! INCORRECT STACK PTR
007136  005216                          INC     (USP)           ;CHECK RESULT
007140  001401                          BEQ     .+4
007142  000000                          HLT                     ;ERROR! INCORRECT RESULT
007144  104000                          SCOPE

                                ;TEST THAT MFPI CAN GET DATA FROM A USER VIRTUAL ADDRESS
                                ;DM=2
007146  012767  170000  170622          MOV     #UM+PUM,PSW     ;USER MODE!!!,PREV USER MODE!!
007154  012766  177777  177776          MOV     #-1,-2(USP)
007162  012702  120000                  MOV     #VIRT,R2        ;R2=VIRTUAL ADDRESS
007166  005037  017200                  CLR     @#PHYS          ;PRESET PHYSICAL ADDRESS
007172  005237  177572                  INC     @#SR0           ;ENABLE MEMORY MANAGEMENT
007176  006522                  UUF2:   MFPI    (R2)+           ;-(USP)=VIRT
007200  005037  177572                  CLR     @#SR0           ;DISABLE MEMORY MANAGEMENT
007204  005716                          TST     (USP)           ;CHECK RESULT
007206  001401                          BEQ     .+4
007210  000000                          HLT                     ;ERROR! INCORRECT RESULT ON STACK
007212  022702  120002                  CMP     #VIRT+2,R2      ;CHECK AUTO INCREMENT
007216  001401                          BEQ     .+4
007220  000000                          HLT                     ;ERROR! AUTO INCREMENT FAILED
007222  005067  170550                  CLR     PSW
007226  104000                          SCOPE

                                ;DM=4
007230  012767  170000  170540          MOV     #UM+PUM,PSW     ;USER MODE!!!,PREV USER MODE!!
007236  012766  177777  177776          MOV     #-1,-2(USP)
007244  012704  120002                  MOV     #VIRT+2,R4      ;R4=VIRTUAL ADDRESS+2
007250  005037  017200                  CLR     @#PHYS          ;PRESET PHYSICAL ADDRESS DATA
007254  005237  177572                  INC     @#SR0           ;ENABLE MEMORY MANAGEMENT
007260  006544                  UUF4:   MFPI    -(R4)           ;-(USP)=VIRT
007262  005037  177572                  CLR     @#SR0           ;DISABLE MEMORY MANAGEMENT
007266  022704  120000                  CMP     #VIRT,R4        ;CHECK AUTO-DECREMENT
007272  001401                          BEQ     .+4
007274  000000                          HLT                     ;ERROR! AUTO-DECREMENT FAILED
007276  005716                          TST     (USP)           ;CHECK RESULT
```

```
007300  001401                        BEQ     .+4
007302  000000                        HLT                     ;ERROR! INCORRECT RESULT
007304  104000                        SCOPE

                            ;DM=6
007306  012767  170000  170462        MOV     #UM+PUM,PSW     ;USER MODE!!!,PREV USER MODE!!
007314  012766  177777  177776        MOV     #-1,-2(USP)
007322  012702  000002                MOV     #2,R2           ;LOAD INDEX REGISTER
007326  005037  017202                CLR     @#PHYS+2        ;PRESET PHYSICAL ADDRESS DATA
007332  005237  177572                INC     @#SR0           ;ENABLE MEMORY MANAGEMENT
007336  006562  120000        UUF6:   MFPI    VIRT(R2)        ;-(USP)-VIRT+2
007342  005037  177572                CLR     @#SR0           ;DISABLE MEMORY MANAGEMENT
007346  022706  000576                CMP     #UPTR-2,USP     ;CHECK STACK PTR
007352  001401                        BEQ     .+4
007354  000000                        HLT                     ;ERROR! INCORRECT STACK PTR
007356  005716                        TST     (USP)           ;CHECK RESULT
007360  001401                        BEQ     .+4
007362  000000                        HLT                     ;ERROR! INCORRECT RESULT
007364  005067  170406                CLR     PSW
007370  104000                        SCOPE

                            ;TEST THAT MFPI OPERATES PROPERLY US PC IN DESTINATION
                            ;DM=3,PC
007372  012767  170000  170376        MOV     #UM+PUM,PSW     ;USER MODE!!!,PREV USER MODE!!
007400  005066  177776                CLR     -2(USP)
007404  012737  177777  017200        MOV     #-1,@#PHYS
007412  005237  177572                INC     @#SR0           ;ENABLE MEMORY MANAGEMENT
007416  006537  120000        UUF11:  MFPI    @#VIRT          ;-(USP)-VIRT
007422  005037  177572                CLR     @#SR0           ;DISABLE MEMORY MANAGEMENT
007426  005216                        INC     (USP)           ;CHECK RESULT
007430  001401                        BEQ     .+4
007432  000000                        HLT                     ;ERROR! INCORRECT RESULT
007434  104000                        SCOPE

                            ;DM=6,PC
007436  012767  170340  170332        MOV     #UM+PUM+PRTY7,PSW       ;USER MODE!!! PREV USER MODE!!
007444  012766  177777  177776        MOV     #-1,-2(USP)
007452  005037  017200                CLR     @#PHYS          ;PRESET PHYSICAL ADDRESS DATA
007456  005237  177572                INC     @#SR0           ;ENABLE MEMORY MANAGEMENT
007462  006567  110312        UUF12:  MFPI    VIRT            ;-(USP)-VIRT
007466  005037  177572                CLR     @#SR0           ;DISABLE MEMORY MANAGEMENT
007472  005716                        TST     (USP)           ;CHECK RESULT
007474  001401                        BEQ     .+4
007476  000000                        HLT                     ;ERROR! INCORRECT RESULT
007500  104000                        SCOPE

                            ;**********************IMPORTANT NOTE**********************
                            ;NO CODE ALLOWED BETWEEN 16600-17776
007502  005067  170270        END:    CLR     PSW
007506  005267  171266                INC     ICNT            ;INCREMENT PASS COUNT
007512  026727  171262  001000        CMP     ICNT,#1000
007520  001402                        BEQ     DONE
007522  000167  171350                JMP     BEGIN
007526  012767  000007  170032  DONE: MOV     #7,TPB          ;RING BELL AFTER 1000
```

```
007534  105767  170024                  TSTB    TPS             ;PASSES
007540  100375                          BPL     .-4
007542  013702  000042                  MOV     @#42,R2         ;MONITOR LOAD?
007546  001405                          BEQ     DONE1           ;NO, CONTINUE
007550  000005                          RESET
007552  004712                          JSR     7,(2)           ;RETURN TO MONITOR
007554  000240                          NOP
007556  000240                          NOP
007560  000240                          NOP
007562  000167  171302          DONE1:  JMP     START           ;RESTART

        000001                          .END
```

```
BEGIN    001076        C      = 000001        DISPLA = 177570        DONE     007526
DONE1    007562        EMTVEC = 000030        END      007502        ERRVEC = 000010
HLT    = 000000        ICNT     001000        KKF0     002604        KKF1     002676
KKF10    003412        KKF11    003502        KKF12    003546        KKF13    003620
KKF14    003666        KKF14A   003670        KKF2     002770        KKF3     003056
KKF4     003122        KKF5     003206        KKF6     003260        KKF7     003350
KK0      001314        KK1      001400        KK10     002104        KK10A    002106
KK10B    002070        KK11     002140        KK11A    002142        KK12     002204
KK13     002270        KK14     002324        KK15     002420        KK16     002456
KK2      001472        KK3      001554        KK4      001614        KK5      001700
KK6      001752        KK7      002046        KM     = 000000        KPAR0  = 172340
KPAR1  = 172342        KPAR2  = 172344        KPAR3  = 172346        KPAR4  = 172350
KPAR5  = 172352        KPAR6  = 172354        KPAR7  = 172356        KPDR0  = 172300
KPDR1  = 172302        KPDR2  = 172304        KPDR3  = 172306        KPDR4  = 172310
KPDR5  = 172312        KPDR6  = 172314        KPDR7  = 172316        KPTR   = 001060
KSP    =%000006        KUF0     005202        KUF1     005302        KUF10    006022
KUF11    006112        KUF12    006156        KUF13    006232        KUF14    006310
KUF2     005374        KUF3     005464        KUF4     005532        KUF5     005616
KUF6     005670        KUF7     005762        KU0      002520        KU1      003740
KU10     004466        KU10A    004472        KU11     004524        KU11A    004532
KU12     004600        KU13     004670        KU13A    004674        KU14     004740
KU15     005036        KU16     005112        KU2      004032        KU3      004120
KU4      004166        KU5      004252        KU6      004324        KU7      004426
MMERR    000462        MMVEC  = 000250        N      = 000010        PC     =%000007
PFVEC  = 000024        PHYS   = 017200        PRTY4  = 000200        PRTY7  = 000340
PSW    = 177776        PUM    = 030000        RW     = 000006        R0     =%000000
R1     =%000001        R2     =%000002        R3     =%000003        R4     =%000004
R5     =%000005        SCOPE  = 104000        SCOPEA   000432        SHLT     000400
SHLTA    000422        SLR    = 177774        SR0    = 177572        SR1    = 177574
SR2    = 177576        SSR0T    001002        START    001070        TEMP     001004
TKB    = 177562        TKS    = 177560        TPB    = 177566        TPS    = 177564
UM     = 140000        UPAR0  = 177640        UPAR1  = 177642        UPAR2  = 177644
UPAR3  = 177646        UPAR4  = 177650        UPAR5  = 177652        UPAR6  = 177654
UPAR7  = 177656        UPDR0  = 177600        UPDR1  = 177602        UPDR2  = 177604
UPDR3  = 177606        UPDR4  = 177610        UPDR5  = 177612        UPDR6  = 177614
UPDR7  = 177616        UPTR   = 000600        USP    =%000006        USRTST   006336
UUF0     007104        UUF11    007416        UUF12    007462        UUF2     007176
UUF4     007260        UUF6     007336        UU0      006402        UU1      006472
UU2      006554        UU3      006632        UU4      006716        UU4A     006722
UU5      006752        UU6      007044        VIRT   = 120000        Z      = 000004
.      = 007566
```

ERRORS DETECTED:  0

```
*MOVV6,MOVV6=MOVV6,SRC
RUN-TIME:  8 14 0 SECONDS
CORE USED:  3K
```

IDENTIFICATION

| | |
|---|---|
| PRODUCT CODE: | MAINDEC-11-DBKTD-A-D |
| PRODUCT NAME: | KT11-D PROCESSOR STATES TEST |
| DATE CREATED: | SEPTEMBER 1, 1972 |
| MAINTAINER: | DIAGNOSTIC GROUP |
| AUTHOR: | R. WHITTON |

COPYRIGHT © 1972
DIGITAL EQUIPMENT CORPORATION

1.Ø ABSTRACT

This is a test that utilizes the KT11-D memory management option and tests that in the two PDP-11/4Ø states (kernel, user) instructions are executed properly. This test tests traps from one state to the other and uses the MFPI/MTPI instructions.

2.Ø REQUIREMENTS

2.1 EQUIPMENT

PDP-11/4Ø with KT11-D (Mem. Mgmt.) installed.

2.2 STORAGE

Utilizes 4K of memory.

3.Ø LOADING PROCEDURE

Load program into memory using absolute loader.

4.Ø STARTING PROCEDURE

Load address 2ØØ. Press start, the program will loop and ring bell on completion of a pass.

5.Ø OPERATION PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

None

5.2 SUBROUTINE ABSTRACTS

5.2.1 SCOPE

Scope is a MOV PC,R1 and store the PC+2 in R1 thus R1 may be used as a reference to determine the last test successfully completed.

5.2.2 HLT

HLT is a halt instruction and is executed whenever a hardware malfunction is detected.

5.3 PROGRAM AND/OR OPERATOR ACTION

5.3.1 PASS COUNT (ICNT)

The number of program passes completed is contained in address ICNT (LOC. 1ØØØ). This address may be examined to determine in which pass the error occured.

## 6.Ø ERRORS

6.1 TEST ERROR WILL CAUSE A HALT

False trap/interrupt errors - the program will halt at the trap vector address +2. The contents of R6 contains the address where the PC of the instruction that caused the trap is stored.

6.2 ERROR RECOVERY

Test errors - Press continue or loop test (see 6.3)
Trap errors - Determine where error occured (see 6.1)

6.3 ERROR LOOPING

To loop on an error replace the HLT instruction with a branch back to the previous scope instruction. Note that if the error is intermittent the test will drop thru the HLT and proceed to the next test. Therefore, to loop the test continuously, replace the BEQ .+4 instruction preceeding the HLT with the branch back to the previous scope.

## 7.Ø RESTRICTIONS

This program must be loaded in lower 4K.

7.1 STARTING RESTRICTION

All programs must be initially started at 2ØØ and may be started at a scope instruction thereafter.

7.2 OPERATIONAL RESTRICTIONS

None

## 8.1 EXECUTION TIME

One pass takes approximately 1Ø seconds.

```
                        ;NLIST  SEQ

                        ;TITLE  TEST DBKTDA PDP11/40 PROCESSER STATES TEST
                        ;ABS

                    ;TEST DBKTDA TESTS FEATURES OF THE TWO PROCESSER STATES AND INCLUDES
                    ;TRAPS FROM ALL STATES TO ALL OTHER STATES, AND MFP/MTP INSTRUCTIONS IN ALL
                    ;STATES AND PREVIOUS STATES.
                    ;NOTE! ALL TESTS ARE ENTERED AND EXITED IN KERNEL MODE;

                    ;STARTING PROCEEDURE
                    ;       LOAD ADDRESS=200
                    ;       PRESS START
                    ;       KERNEL STACK POINTER IS AT 500
                    ;       USER STACK POINTER IS AT 700
                    ;       BELL WILL RING WHEN TEST IS COMPLETE

                    ;REGISTER ASSIGNMENTS
000000              R0=%0
000001              R1=%1
000002              R2=%2
000003              R3=%3
000004              R4=%4
000005              R5=%5
000007              PC=%7

                    ;STACK POINTERS
000006                      KSP=%6                  ;KERNEL STACK POINTER
000006                      USP=%6                  ;USER STACK POINTER
000008                      HLT=HALT
010701              SCOPE=010701                    ;MOVE PC TO R1
000003                      TRT=3                   ;TRACE TRAP
000140                      PRTY3=140
000200                      PRTY4=200
000340                      PRTY7=340

                    ;VECTOR ADDRESSES
000004                      ERRVEC=4                ;ADDRESS OF ERROR VECTOR
000010                      RESVEC=10               ;ADDRESS OF RESERVED INST TRAP VECTOR
000030                      EMTVEC=30               ;ADDRESS OF EMT VECTOR
000034                      TRAPVEC=34              ;ADDRESS OF TRAP VECTOR
000020                      IOTVEC=20               ;ADDRESS OF IOT VECTOR
000014                      TBITVEC=14              ;ADDRESS OF 'T' BIT TRAP VECTOR
000014                      TRTVEC=14               ;ADDRESS OF 'TRACE' TRAP
000064                      TPVEC=64                ;ADDRESS OF TTY PRINTER INTERRUPT VECTOR
                    ;HARDWARE REGISTER ASSIGNMENTS
177776                      PSW=177776              ;ADDRESS OF STATUS REGISTER
177774                      SLR=177774              ;ADDRESS OF STACK LIMIT REGISTER
177560                      TKS=177560              ;ADDRESS OF KEYBOARD CSR
177562                      TKB=177562              ;ADDRESS OF KEYBOARD BUFFER
177564                      TPS=177564              ;ADDRESS OF TELEPRINTER CSR
177566                      TPB=177566              ;ADDRESS OF TELEPRINTER BUFFER
177570                      SWR=177570              ;ADDRESS OF CONSOL SWITCH REGISTER
                    ;INITIAL STACK POINTER SETTIGS
```

```
                000500                  KPTR=500                ;KERNEL INITIAL STACK POINTER VALUE
                000700                  UPTR=700                ;USER INITIAL STACK POINTER VALUE
                001000                  YELPTR=1000             ;STACK POINTER VALUE FOR 'YELLOW' OVFLW
                000736                  REDPTR=736              ;STACK POINTER VALUE FOR 'RED' OVFLW

                                ;MISC, BIT ASSIGNMENTS
                100000                  BIT15=100000
                040000                  BIT14=40000
                020000                  BIT13=20000
                000100                  BIT6=100


                                ;STATUS REGISTER BIT ASSIGNMENTS
                140000                  UM=140000               ;USER MODE
                100000                  IM=100000               ;ILLEGAL MODE
                040000                  IM1=40000               ;ILLEGAL MODE
                000000                  KM=0                    ;KERNEL MODE
                030000                  PUM=030000              ;PREVIOUS USER MODE
                000000                  PKM=0                   ;PREVIOUS KERNEL MODE
                000000                  REG=0                   ;REGISTER BIT  HAS NO EFFECT!!!!
                000020                  TBIT=20                 ;'T' BIT IN P#PSW
                000001                  C=1                     ;'C' BIT IN PS
                000002                  V=2                     ;'V' BIT IN PS
                000004                  Z=4                     ;'Z' BIT IN PS
                000010                  N=10                    ;'N' BIT IN PS


                000000                  .=0
                                        .REPT   100
                                        .+2
                                        HALT
                                        .ENDR

                000200                  .=200
000200  000167  000606                  JMP     START           ;GO START

                001000                  .=1000
                                ;TAGS

001000  000000                  ICNT:  0                        ;CONTAINS PASS COUNT
001002  000000                  TEMP:  0
                001012                  .=.+6
```

```
001012  012706  000500          START:  MOV     #KPTR,KSP
001016  005067  177756                  CLR     ICNT
                                ;TEST THAT PROCESSER POWERED UP OK FOR THE TEST
001022  032737  000000  177776  PWRUP:  BIT     #KM+PKM,@#PSW   ;IS STATUS CORRECT
001030  001377                          BNE     .               ;LOOP HERE IF NOT

001032  012706  000500          BEGIN:  MOV     #KPTR,KSP       ;INITIALIZE THE STACK POINTER

                                ;CHECK THAT THE NOP INSTRUCTION IS A 'NOP' IN USER MODE.
001036  010701                  T1:     SCOPE
001040  012737  140000  177776          MOV     #UM,@#PSW       ;USER MODE,PRIORITY LEVEL 0
001046  000240                          NOP
001050  013700  177776                  MOV     @#PSW,R0        ;GET @#PSW
001054  005037  177776                  CLR     @#PSW           ;KERNEL MODE!!!
001060  022700  140000                  CMP     #UM,R0          ;TEST THAT NOP DID NOT ALTER @#PSW
001064  001401                          BEQ     .+4
001066  000000                          HLT                     ;ERROR! NOP CHANGED STATUS WORD


                                ;TEST TRAP FROM USER MODE TO KERNEL MODE
001070  010701                  T5:     SCOPE
001072  012706  000500                  MOV     #KPTR,KSP
001076  012737  001134  000020          MOV     #T5A,@#IOTVEC
001104  005067  176712                  CLR     IOTVEC+2
001110  012737  140340  177776          MOV     #UM+PRTY7,@#PSW ;USER MODE!!!
001116  012706  000700                  MOV     #UPTR,USP
001122  000277                          SCC
001124  000004                          IOT
001126  005037  177776          T5AA:   CLR     @#PSW
001132  000000                          HLT
001134  013700  177776          T5A:    MOV     @#PSW,R0
001140  005037  177776                  CLR     @#PSW
001144  022700  030000                  CMP     #KM+PUM,R0
001150  001401                          BEQ     .+4
001152  000000                          HLT
001154  022767  001126  177312          CMP     #T5AA,KPTR+4
001162  001401                          BEQ     .+4
001164  000000                          HLT
001166  022767  140357  177302          CMP     #UM+PRTY7+17,KPTR+2
001174  001401                          BEQ     .+4
001176  000000                          HLT
001200  022706  000474                  CMP     #KPTR+4,KSP
001204  001401                          BEQ     .+4
001206  000000                          HLT
001210  012737  140000  177776          MOV     #UM,@#PSW
001216  010600                          MOV     USP,R0
001220  005037  177776                  CLR     @#PSW
001224  022700  000700                  CMP     #UPTR,R0
001230  001401                          BEQ     .+4
001232  000000                          HLT
001234  012737  000022  000020          MOV     #IOTVEC+2,@#IOTVEC

                                ;TEST TRAP FROM USER TO USER MODE (VIA TRACE TRAP)
```

```
001242  010701                          T7:     SCOPE
001244  012767  001302  176542                  MOV     #T7A,TRTVEC
001252  012767  140000  176536                  MOV     #UM,TRTVEC+2    ;USER MODE ON TRAP
001260  012737  140000  177776                  MOV     #UM,@#PSW
001266  012706  000700                          MOV     #UPTR,USP
001272  000003                                  TRT
001274  005037  177776          T7AA:   CLR     @#PSW
001300  000000                                  HLT
001302  013700  177776          T7A:    MOV     @#PSW,R0
001306  010602                                  MOV     USP,R2
001310  042737  140000  177776                  BIC     #UM,@#PSW
001316  022767  001274  177350                  CMP     #T7AA,UPTR+4
001324  001401                                  BEQ     .+4
001326  000000                                  HLT
001330  022700  170000                          CMP     #UM+PUM,R0
001334  001401                                  BEQ     .+4
001336  000000                                  HLT
001340  012767  000016  176446                  MOV     #TRTVEC+2,TRTVEC
001346  005067  176444                          CLR     TRTVEC+2

                                        ;TEST THAT THE 'HALT' INSTRUCTION TRAPS TO LOCATION 10 IN
                                        ;USER MODE.
001352  010701                          T12:    SCOPE
001354  012737  001410  000010                  MOV     #T12A,@#RESVEC
001362  005037  000012                          CLR     @#RESVEC+2
001366  012706  000500                          MOV     #KPTR,KSP
001372  012737  140000  177776                  MOV     #UM,@#PSW       ;USER MODE!!!
001400  000000                                  HALT                    ;HALT TRAPS IN USER MODE
001402  005037  177776          T12AA:  CLR     @#PSW
001406  000000                                  HALT                    ;ERROR!  HALT DID NOT TRAP
001410  013700  177776          T12A:   MOV     @#PSW,R0
001414  005037  177776                  CLR     @#PSW
001420  022700  030000                  CMP     #KM+PUM,R0
001424  001401                          BEQ     .+4
001426  000000                          HLT
001430  022767  001402  177036                  CMP     #T12AA,KPTR+4
001436  001401                                  BEQ     .+4
001440  000000                                  HLT

                                        ;CHECK THAT SPL TRAPS TO 10 IN USER MODE.
001442  010701                          T13:    SCOPE
001444  012737  001474  000010                  MOV     #T13A,@#RESVEC
001452  012706  000500                          MOV     #KPTR,KSP       ;SET KERNEL STACK PTR
001456  012737  140000  177776                  MOV     #UM,@#PSW       ;USER MODE!!!
001464  000237                                  SPL     7               ;SPL TRAPS IN USER MODE
001466  005037  177776          T13AA:  CLR     @#PSW           ;KERNEL MODE!!!
001472  000000                                  HLT                     ;ERROR! SPL FAILED TO TRAP IN USER MODE
001474  013700  177776          T13A:   MOV     @#PSW,R0
001500  005037  177776                  CLR     @#PSW
001504  022700  030000                  CMP     #KM+PUM,R0
001510  001401                          BEQ     .+4
001512  000000                          HLT
001514  022767  001466  176752                  CMP     #T13AA,KPTR+4
001522  001401                                  BEQ     .+4
```

```
001524  000000                          HLT
001526  012737  000012  000010          MOV     #RESVEC+2,@#RESVEC

                                ;TEST THAT "RESET" RESETS IN KERNEL MODE
001534  010701                  T18:    SCOPE
001536  005037  177776                  CLR     @#PSW
001542  012737  000340  177776          MOV     #PRTY7,@#PSW    ;PRIORITY TO 7
001550  012767  000100  176006          MOV     #100,177564     ;SET "IE" IN TPS
001556  000005                          RESET                   ;CLEAR "IE"
001560  005037  177776                  CLR     @#PSW
001564  032767  000100  175772          BIT     #100,177564
001572  001401                          BEQ     .+4
001574  000000                          HLT                     ;RESET DID NOT
                                                                ;CLEAR "IE"

                                ;TEST THAT "RESET" NOP'S IN USER MODE
001576  010701                  T19:    SCOPE
001600  012737  140340  177776          MOV     #UM+PRTY7,@#PSW ;USER MODE!!!
001606  012767  000100  175750          MOV     #100,177564     ;SET "IE"
001614  000005                          RESET                   ;SHOULD NOP
001616  032767  000100  175740          BIT     #100,177564
001624  001001                          BNE     .+4
001626  000000                          HLT                     ;"IE" CLEARED
001630  005067  175730                  CLR     177564
001634  005037  177776                  CLR     @#PSW

                                ;TEST INTERRUPT SEQUENCE USER TO KERNEL MODE
001640  010701                  T1S:    SCOPE
001642  012706  000500                  MOV     #KPTR,KSP       ;SET KERNEL STACK POINTER
001646  012737  170340  177776          MOV     #UM+PUM+PRTY7,@#PSW     ;USER MODE!!!
001654  012767  001720  176202          MOV     #T15A,64                ;INTERRUPT VEC',
001662  012767  000200  176176          MOV     #KM+PRTY4,66
001670  012706  000700                  MOV     #UPTR,USP       ;SET USER STACK POINTER
001674  042737  000200  177776          BIC     #PRTY4,@#PSW    ;SET PRIORITY LEVEL=3
001702  012767  000100  175654          MOV     #100,177564     ;REQUEST AN INTERRUPT AT LEVEL 4
001710  000240                          NOP
001712  005037  177776          T15AA:  CLR     @#PSW           ;KERNEL MODE!!!
001716  000000                          HLT                     ;ERROR! NO INTERRUPT REQUEST
001720  013700  177776          T15A:   MOV     @#PSW,R0        ;GET 'NEW' @#PSW
001724  005067  175634                  CLR     177564          ;DISABLE REQUEST
001730  005037  177776                  CLR     @#PSW
001734  022700  030200                  CMP     #KM+PUM+PRTY4,R0  ;TEST THAT 'NEW' @#PSW IS CORRECT
001740  001401                          BEQ     .+4             ;(PIRVEC+2)
001742  000000                          HLT                     ;ERROR! 'NEW' @#PSW NOT = TO (PIRVEC+2)
001744  022767  001712  176522          CMP     #T15AA,KPTR-4   ;IS RETURN ADDRESS ON KERNEL STACK
001752  001401                          BEQ     .+4
001754  000000                          HLT                     ;ERROR! RETURN ADDRESS NOT ON KERNEL STACK
001756  022767  170140  176512          CMP     #UM+PUM+PRTY3,KPTR+2  ;TEST THAT 'OLD' @#PSW WAS SAVED ON
001764  001401                          BEQ     .+4             ;KERNEL STACK
001766  000000                          HLT                     ;ERROR!
001770  012767  000066  176066          MOV     #66,64
001776  005067  176064                  CLR     66

                                ;TEST THAT THERE IS NO STACK OVERFLOW IN USER MODE',
```

```
002002  010701                  T17:    SCOPE
002004  012737  000400  177774          MOV     #400,@#SLR      ;SET STACK LIMIT =1000
002012  012737  140000  177776          MOV     #UM,@#PSW       ;USER MODE!!!
002020  012737  002250  000004          MOV     #T17ERR,@#ERRVEC
002026  012706  000700                  MOV     #UPTR,USP       ;SET USER STACK POINTER
002032  005067  176744                  CLR     TEMP            ;CLEAR INDICATOR LOCATION
002036  004767  000006          T17A:   JSR     7,T17B          ;PUSH ONTO USER STACK
002042  052767  000400  176732          BIS     #400,TEMP       ;SET ERROR INDICATOR BIT
002050  052767  000001  176724  T17B:   BIS     #1,TEMP         ;SET INDICATOR BIT
002056  004567  000006                  JSR     5,T17C          ;PUSH ONTO USER STACK
002062  052767  001000  176712          BIS     #1000,TEMP      ;SET ERRROR INDICATOR BIT
002070  052767  000002  176704  T17C:   BIS     #2,TEMP         ;SET INDICATOR BIT
002076  050546                          BIS     R5,-(USP)       ;PUSH ONTO USER STACK
002100  052767  000004  176674          BIS     #4,TEMP         ;SET INDICATOR BIT
002106  052737  000000  177776          BIS     #REG,@#PSW      ;SELECT R0-R5
002114  004767  000006                  JSR     7,T17D          ;PUSH ONTO USER STACK
002120  052767  002000  176654          BIS     #2000,TEMP      ;SET ERROR INDICATOR BIT
002126  052767  000010  176646  T17D:   BIS     #10,TEMP
002134  012702  002150                  MOV     #T17E,R2        ;SET UP RETURN FOR RTS
002140  000202                          RTS     R2              ;GO TO T16E
002142  052767  004000  176632          BIS     #4000,TEMP      ;SET INDICATOR TO SHOW ERROR
002150  052767  000020  176624  T17E:   BIS     #20,TEMP
002156  004567  000006                  JSR     R5,T17F
002162  052767  010000  176612          BIS     #10000,TEMP     ;SET ERROR INDICATOR BIT
002170  052767  000040  176604  T17F:   BIS     #40,TEMP
002176  012737  002222  000034          MOV     #T17G,@#TRAPVEC ;SET UP TRAP VECTOR FOR TRAP
002204  012737  140000  000036          MOV     #UM,@#TRAPVEC+2
002212  104400                          TRAP
002214  052767  020000  176560          BIS     #20000,TEMP
002222  052767  000100  176552  T17G:   BIS     #100,TEMP
002230  005037  177776                  CLR     @#PSW           ;KERNEL MODE!!!
002234  022767  000177  176540          CMP     #177,TEMP
002242  001401                          BEQ     .+4
002244  000000                          HLT
002246  000403                          BR      T17X
002250  005037  177776  T17ERR: CLR     @#PSW
002254  000000                          HLT                     ;ERROR: OVERFLOW OCCURED
002256  005037  177774  T17X:   CLR     @#SLR
002262  012737  000036  000034          MOV     #TRAPVEC+2,@#TRAPVEC
002270  005067  175542                  CLR     TRAPVEC+2

                                ;TEST THAT MTPD/I POPS WORD OFF THE THE APPROPRIATE STACK (AS
                                ;DETERMINED BY BITS 15&14 IN @#PSW.)
                                ;MTPD, KERNEL MODE
002274  010701                  T21:    SCOPE
002276  005037  177776                  CLR     @#PSW
002302  012706  000500                  MOV     #KPTR,KSP               ;SET KERNEL STACK POINTER
002306  012700  177777                  MOV     #-1,R0                  ;PRE-SET R0
002312  005016                          CLR     (KSP)                   ;PUT 0 ON THE STACK
002314  012737  030011  177776          MOV     #PUM+N+C,@#PSW          ;PRE SET STATUS
002322  006600                          MTPI    R0                      ;R0<--(KSP)+

002324  013702  177776                  MOV     @#PSW,R2                ;GET STATUS
002330  022702  030005                  CMP     #PUM+Z+C,R2
```

```
002334  001401                  BEQ     .+4
002336  000000                  HLT
002340  022706  000502          CMP     #KPTR+2,KSP             ;ERROR: INCORRECT STATUS
002344  001401                  BEQ     .+4                     ;DID KSP INCREMENT BY 2
002346  000000                  HLT
002350  005700                  TST     R0                      ;ERROR:  KSP DID NOT POP
002352  001401                  BEQ     .+4                     ;DID WORD ON STACK (0) GET TO R0?
002354  000000                  HLT

                                                                ;ERROR: MTPD DID NOT POP 0 OFF
                                                                ;KSP INTO R0

                        ;MTPI,KERNEL MODE
002356  010701          T22:    SCOPE
002360  005037  177776          CLR     @#PSW
002364  012706  000500          MOV     #KPTR,KSP
002370  005002                  CLR     R2                      ;PRESET R2
002372  012716  177777          MOV     #-1,(KSP)
002376  012737  030006  177776  MOV     #PUM+Z+V,@#PSW          ;PRESET STATUS
002404  006602                  MTPI    R2                      ;R2=(KSP)+

002406  013700  177776          MOV     @#PSW,R0                ;GET STATUS
002412  022700  030010          CMP     #PUM+N,R0
002416  001401                  BEQ     .+4
002420  000000                  HLT                             ;ERROR: INCORRECT STATUS
002422  022706  000502          CMP     #KPTR+2,KSP
002426  001401                  BEQ     .+4
002430  000000                  HLT                             ;ERROR:
002432  005202                  INC     R2
002434  001401                  BEQ     .+4
002436  000000                  HLT                             ;ERROR:

                        ;MTPD, USER MODE
002440  010701          T25:    SCOPE
002442  012737  140000  177776  MOV     #UM,@#PSW
002450  012706  000700          MOV     #UPTR,USP
002454  052716  177777          BIS     #-1,(USP)
002460  000261                  SEC
002462  042705  177777          BIC     #-1,R5
002466  006605                  MTPI    R5                      ;R5=(USP)+

002470  013700  177776          MOV     @#PSW,R0
002474  010602                  MOV     USP,R2
002476  005037  177776          CLR     @#PSW
002502  022700  140011          CMP     #UM+N+C,R0
002506  001401                  BEQ     .+4
002510  000000                  HLT
002512  022702  000702          CMP     #UPTR+2,R2
002516  001401                  BEQ     .+4
002520  000000                  HLT
002522  005205                  INC     R5
002524  001401                  BEQ     .+4
002526  000000                  HLT


                        ;MTPI, USER MODE
```

```
002530  010701                      T261:   SCOPE
002532  012737  140000  177776              MOV     #UM,@#PSW
002540  012706  000700                      MOV     #UPTR,USP
002544  042716  177777                      BIC     #-1,(USP)
002550  052700  177777                      BIS     #-1,R0
002554  000257                              CCC
002556  006600                              MTPI    R0                      ;R0<-(USP)+

002560  013702  177776                      MOV     @#PSW,R2
002564  010603                              MOV     USP,R3

002566  005037  177776                      CLR     @#PSW
002572  022702  140004                      CMP     #UM+2,R2
002576  001401                              BEQ     .+4
002600  000000                              HLT
002602  022703  000702                      CMP     #UPTR+2,R3
002606  001401                              BEQ     .+4
002610  000000                              HLT
002612  005700                              TST     R0
002614  001401                              BEQ     .+4
002616  000000                              HLT

                                    ;TEST THAT MTP D/I POPS WORD OFF STACK (AS DETERMINED BY BITS 15 & 14
                                    ;INTO STACK POINTER (AS DETERMINED BY BITS 13 &12);
                                    ;USP<-(KSP)+,MTPD
002620  010701                      T301:   SCOPE
002622  012737  140000  177776              MOV     #UM,@#PSW               ;USER MODE!!!
002630  005006                              CLR     USP                     ;PRESET USER STACK POINTER
002632  012737  030000  177776              MOV     #KM+PUM,@#PSW           ;KERNEL MODE!!!, PREV USER MODE!!
002640  012706  000500                      MOV     #KPTR,KSP               ;SET KERNEL STACK POINTER
002644  012716  000700                      MOV     #UPTR,(KSP)
002650  000277                              SCC                             ;PRESET CC'S
002652  006606                              MTPI    USP                     ;USP<-(KSP)+

002654  013702  177776                      MOV     @#PSW,R2                ;SAVE CC'S
002660  012737  140000  177776              MOV     #UM,@#PSW               ;USER MODE!!!
002666  010600                              MOV     USP,R0                  ;GET USER STACK POINTER
002670  005037  177776                      CLR     @#PSW                   ;KERNEL MODE!!!
002674  022700  000700                      CMP     #UPTR,R0                ;CHECK THAT MTPD SET USER STACK
002700  001401                              BEQ     .+4                     ;POINTER PROPERLY
002702  000000                              HLT                             ;ERROR!
002704  022706  000502                      CMP     #KPTR+2,KSP             ;CHECK KERNEL STACK POINTER
002710  001401                              BEQ     .+4
002712  000000                              HLT



                                    ;KSP<-(KSP)+,MTPD
002714  010701                      T311:   SCOPE
002716  012706  000500                      MOV     #KPTR,KSP
002722  012716  000736                      MOV     #REDPTR,(KSP)
002726  006606                              MTPI    KSP                     ;KSP<-(KSP)+
002730  022706  000736                      CMP     #REDPTR,KSP
002734  001401                              BEQ     .+4
002736  000000                              HLT
```

```
                                        ;;USP<-(USP)+,MTPD
002740  010701                          T31C:   SCOPE
002742  012737  170000  177776                  MOV     #UM+PUM,@#PSW   ;USER MODE!!!, PREV USER MODE!!
002750  012706  000700                          MOV     #UPTR,USP       ;SET USER STACK PTR
002754  005016                                  CLR     (USP)           ;PUT #0 ON USER STACK
002756  000257                                  CCC
002760  006606                                  MTPI    USP             ;USP<-(USP)+

002762  013700  177776                          MOV     @#PSW,R0                ;SAVE CC'S
002766  010602                                  MOV     USP,R2          ;SAVE USER STACK POINTER
002770  005037  177776                          CLR     @#PSW           ;KERNEL MODE!!!
002774  022700  170004                          CMP     #UM+PUM+Z,R0    ;CHECK STATUS
003000  001401                                  BEQ     .+4
003002  000000                                  HLT                     ;ERROR! INCORRECT STATUS AFTER MTPD
003004  005702                                  TST     R2              ;CHECK NEW STACK POINTER VALUE
003006  001401                                  BEQ     .+4
003010  000000                                  HLT                     ;ERROR! MTPD FAILED TO SET USER STACK POINTER

                                        ;USP<-(KSP)+,MTPI
003012  010701                          T32A:   SCOPE
003014  012737  140000  177776                  MOV     #UM,@#PSW       ;USER MODE
003022  012706  177777                          MOV     #-1,USP         ;PRESET USER STACK POINTER
003026  012737  030000  177776                  MOV     #KM+PUM,@#PSW   ;CURRENT KERNEL,PREVIOUS USER
003034  005046                                  CLR     -(KSP)
003036  006606                                  MTPI    USP             ;USP<-(KSP+

003040  012737  140000  177776                  MOV     #UM,@#PSW
003046  010600                                  MOV     USP,R0                  ;GET USER STACK POINTER
003050  005037  177776                          CLR     @#PSW
003054  005700                                  TST     R0
003056  001401                                  BEQ     .+4
003060  000000                                  HLT

                                        ;USP<-(USP)+
003062  010701                          T35:    SCOPE
003064  012737  170000  177776                  MOV     #UM+PUM,@#PSW
003072  012706  000700                          MOV     #UPTR,USP
003076  012716  000700                          MOV     #UPTR,(USP)
003102  006606                                  MTPI    USP                     ;USP<-(USP)+

003104  010600                                  MOV     USP,R0
003106  005037  177776                          CLR     @#PSW
003112  022700  000700                          CMP     #UPTR,R0
003116  001401                                  BEQ     .+4
003120  000000                                  HLT


                                        ;TEST THAT MTPD/I TRAPS ON AN ODD ADDRESS DESTINATION
                                        ;KERNEL MODE
003122  010701                          T36:    SCOPE
003124  005037  177776                          CLR     @#PSW
003130  012706  000500                          MOV     #KPTR,KSP
003134  012716  177777                          MOV     #-1,(KSP)
```

```
003140  012737  003160  000004          MOV     #T36A,@#ERRVEC
003146  005067  174634                  CLR     ERRVEC+2
003152  006667  174621                  MTPI    -1                      ;TRAPS ON ODD ADDRESS
003156  000000                  T36AA:  HLT                             ;ERROR: DID NOT TRAP
003160  022706  000476          T36A:   CMP     #KPTR-2,KSP             ;IS KSP CORRECT?(1 POP AND 2
003164  001401                          BEQ     .+4                     ;PUSHES)
003166  000000                          HLT                             ;ERROR: INCORRECT VALUE IN KSP
003170  022767  003156  175300          CMP     #T36AA,KPTR-2
003176  001401                          BEQ     .+4
003200  000000                          HLT

                                ;USER MODE
003202  010701                  T40:    SCOPE
003204  012737  170000  177776          MOV     #UM+PUM,@#PSW           ;USER MODE!!!, PREV USER MODE!!
003212  012702  000001                  MOV     #1,R2
003216  012706  000700                  MOV     #UPTR,USP               ;SET USER STACK POINTER
003222  012716  125252                  MOV     #125252,(USP)           ;PRESET USER STACK
003226  012737  003252  000004          MOV     #T40A,@#ERRVEC          ;LOAD ERROR VECTOR
003234  012737  140000  000006          MOV     #UM,@#ERRVEC+2
003242  006642                          MTPI    -(R2)                   ;-(R2)+(USP)+;SHOULD TRAP ON ODD ADRS
003244  005037  177776          T40AA:  CLR     @#PSW                   ;KERNEL MODE!!!
003250  000000                          HLT                             ;ERROR DID NOT TRAP
003252  010600                  T40A:   MOV     USP,R0                  ;GET USERS STACK POINTER
003254  042737  140000  177776          BIC     #UM,@#PSW               ;KERNEL MODE!!!
003262  022700  000676                  CMP     #UPTR-2,R0              ;CHECK THAT USER STACK POINTER
003266  001401                          BEQ     .+4                     ;PUSHED PROPERLY (1 POP,2 PUSHES)
003270  000000                          HLT                             ;ERROR: INCORRECT USER STACK POINTER
003272  022737  170010  000700          CMP     #UM+PUM+N,@#UPTR                ;CHECK THAT CORRECT STATUS WAS
003300  001401                          BEQ     .+4                     ;SAVED ON USER STACK ('N' IS DATA POPPED)
003302  000000                          HLT                             ;ERROR: INCORRECT STATUS SAVED ON USER STACK
003304  022767  003244  175364          CMP     #T40AA,UPTR-2           ;CHECK THAT RETURN ADDRESS WAS
003312  001401                          BEQ     .+4                     ;SAVED ON USER STACK
003314  000000                          HLT                             ;ERROR: RETURN PC NOT ON USER STACK
003316  022702  177777                  CMP     #-1,R2                          ;DID R2 DECREMENT BY 2
003322  001401                          BEQ     .+4
003324  000000                          HLT
                                ;TEST THAT MTP D/I CAN LOAD MEMORY ADDRESSES,
                                ;KERNEL MODE
003326  010701                  T41:    SCOPE
003330  005037  177776                  CLR     @#PSW
003334  012700  177777                  MOV     #-1,R0
003340  012737  003374  000004          MOV     #T41A,@#ERRVEC
003346  005067  174434                  CLR     ERRVEC+2
003352  052737  000000  177776          BIS     #REG,@#PSW              ;R0-R5
003360  005000                          CLR     R0
003362  012746  000002                  MOV     #2,-(KSP)
003366  000261                          SEC
003370  006620                          MTPI    (R0)+                           ;(R0)+=(KSP)+
003372  000401                          BR      .+4
003374  000000                  T41A:   HLT                             ;ERROR: TRAPPED
003376  103401                          BCS     .+4                     ;MTP D/I SHOULD NOT AFFECT CARRY
003400  000000                          HLT                             ;BIT ERROR: CARRY BIT BUT CLEARED;
003402  022767  000002  174370          CMP     #2,0
003410  001401                          BEQ     .+4
```

```
003412  000000                          HLT


003414  010701                  T41B:   SCOPE
003416  012737  003444  000004          MOV     #T41BB,@#ERRVEC ;LOAD ERROR VECTOR
003424  012706  000500                  MOV     #KPTR,KSP       ;SET KERNEL STACK POINTER
003430  012716  177777                  MOV     #-1,(KSP)       ;LOAD KERNEL STACK
003434  000257                          CCC                     ;PRESET CC'S
003436  006637  001002                  MTPI    @#TEMP          ;@#TEMP-(KSP)-

003442  000401                          BR      .+4
003444  000000                  T41BB:  HLT                     ;ERROR! TRAPPED
003446  013700  177776                  MOV     @#PSW,R0                ;SAVE CC'S
003452  022700  000010                  CMP     #REG+N,R0       ;CHECK RESULT STATUS
003456  001401                          BEQ     .+4
003460  000000                          HLT                     ;ERROR! INCORRECT STATUS AFTER MTPD
003462  005237  001002                  INC     @#TEMP          ;CHECK RESULT
003466  001401                          BEQ     .+4
003470  000000                          HLT                     ;ERROR! MTPD FAILED

                                ;USER MODE
003472  010701                  T43:    SCOPE
003474  005037  177776                  CLR     @#PSW
003500  012703  177777                  MOV     #-1,R3
003504  012737  003544  000004          MOV     #T43A,@#ERRVEC
003512  012737  140000  177776          MOV     #UM,@#PSW
003520  012703  001004                  MOV     #TEMP+2,R3
003524  005067  175252                  CLR     TEMP
003530  012706  000700                  MOV     #UPTR,USP
003534  052716  177777                  BIS     #-1,(USP)
003540  006643                          MTPI    -(R3)                   ;-(R3)-(USP)-
003542  000401                          BR      .+4
003544  000000                  T43A:   HLT                     ;ERROR TRAPPED
003546  013700  177776                  MOV     @#PSW,R0
003552  042737  140000  177776          BIC     #UM,@#PSW               ;KERNEL MODE!!!
003560  122700  000010                  CMPB    #N,R0
003564  001401                          BEQ     .+4
003566  000000                          HLT
003570  005167  175206                  COM     TEMP
003574  001401                          BEQ     .+4
003576  000000                          HLT
003600  012737  000006  000004          MOV     #ERRVEC+2,@#ERRVEC
003606  005067  174174                  CLR     ERRVEC+2

                                ;TEST THAT MFP D/I PUSHES DESTINATION REGISTER DATA ONTO THE APPROPRIATE STACK
                                ;(AS DETERMINED BY @#PSW BITS 15 &14)
                                ;KERNEL MODE,MFPD
003612  010701                  T44:    SCOPE
003614  012706  000500                  MOV     #KPTR,KSP
003620  012716  125252                  MOV     #125252,(KSP)
003624  012700  177777                  MOV     #-1,R0
003630  000261                          SEC
003632  006500                          MFPI    R0                      ;-(KSP)-R0,(R0)=-1
003634  013702  177776                  MOV     @#PSW,R2                ;GET STATUS RESULT
```

```
003640  022702  000011                CMP     #REG+N+C,R2     ;
003644  001401                        BEQ     .+4
003646  000000                        HLT                     ;ERROR! INCORRECT STATUS RESULT
003650  022706  000476                CMP     #KPTR+2,KSP     ;DID KERNEL STACK POINTER GET
003654  001401                        BEQ     .+4             ;PUSHED?
003656  000000                        HLT                     ;ERROR!
003660  005116                        COM     (KSP)           ;TEST THAT CORRECT DATA(=1) GOT
003662  001401                        BEQ     .+4             ;PUSHED ONTO KERNEL STACK
003664  000000                        HLT                     ;ERROR! =1NOT PUSHED ONTO KERNEL STACK
                              ;KERNEL MODE,MFPI
003666  010701                T451    SCOPE
003670  012706  000500                MOV     #KPTR,KSP
003674  012716  052525                MOV     #52525,(KSP)    ;PRE SET STACK
003700  005004                        CLR     R4              ;PRESET (WRONG) REGISTER
003702  012737  000001  177776        MOV     #REG+C,@#PSW    ;SELECT R0=R5,SET C
003710  012704  125252                MOV     #125252,R4      ;LOAD DATA TO BE MOVED
003714  006504                        MFPI    R4              ;=(KSP)=R4,(R4)=125252

003716  013700  177776                MOV     @#PSW,R0
003722  022700  000011                CMP     #REG+N+C,R0     ;CHECK STATUS RESULT
003726  001401                        BEQ     .+4
003730  000000                        HLT                     ;ERROR! INCORRECT STATUS
003732  022706  000476                CMP     #KPTR+2,KSP     ;CHECK PUSH
003736  001401                        BEQ     .+4
003740  000000                        HLT                     ;ERROR! KSP DID NOT PUSH DOWN
003742  022716  125252                CMP     #125252,(KSP)   ;CHECK DATA ON THE STACK
003746  001401                        BEQ     .+4
003750  000000                        HLT                     ;ERROR! INCORRECT DATA ON THE STACK
                                                              ;IF DATA=0 THEN INCORRECT REGISTER
                                                              ;(R4), IF DATA=52525 NO DATA PUSHED
                                                              ;ON THE STACK;
                              ;USER MODE,MFPD
003752  010701                T501    SCOPE
003754  005003                        CLR     R3              ;PRESET
003756  012737  140000  177776        MOV     #UM,@#PSW       ;USER MODE, R0=R5
003764  012706  000700                MOV     #UPTR,USP       ;SET USER'S STACK POINTER
003770  012726  125252                MOV     #125252,(USP)+  ;PRESET STACK
003774  012703  177777                MOV     #-1,R3          ;
004000  000257                        CCC
004002  006503                        MFPI    R3              ;=(USP)=R3 (R3)=-1

004004  013700  177776                MOV     @#PSW,R0
004010  010604                        MOV     USP,R4
004012  042737  140000  177776        BIC     #UM,@#PSW
004020  022700  140010                CMP     #UM+N,R0
004024  001401                        BEQ     .+4
004026  000000                        HLT
004030  022704  000700                CMP     #UPTR,R4
004034  001401                        BEQ     .+4
004036  000000                        HLT
004040  005214                        INC     (R4)
004042  001401                        BEQ     .+4
004044  000000                        HLT
004046  005037  177776                CLR     @#PSW
```

```
                                        ;USER MODE,MFPI
004052  010701                          T51:    SCOPE
004054  005005                                  CLR     R5
004056  012737  140000  177776                  MOV     #UM,@#PSW       ;USER MODE!!!
004064  012706  000700                          MOV     #UPTR,USP       ;SET USER STACK POINTER
004070  012716  177777                          MOV     #-1,(USP)       ;PRESET USER STACK
004074  012705  000700                          MOV     #UPTR,R5        ;PRESET R5
004100  000277                                  SCC                     ;PRESET CONDITION CODES
004102  006505                                  MFPI    R5              ;-(USP)-R5

004104  013700  177776                          MOV     @#PSW,R0                ;GET STATUS RESULT
004110  010602                                  MOV     USP,R2          ;GET USER STACK POINTER
004112  042737  140000  177776                  BIC     #UM,@#PSW               ;KERNEL MODE!!!
004120  022700  140001                          CMP     #UM+C,R0        ;CHECK STATUS RESULT AFTER MFPI INST
004124  001401                                  BEQ     .+4
004126  000000                                  HLT                     ;ERROR! INCORRECT STATUS AFTER MFPI
004130  022702  000676                          CMP     #UPTR-2,R2
004134  001401                                  BEQ     .+4
004136  000000                                  HLT
004140  022712  000700                          CMP     #UPTR,(R2)
004144  001401                                  BEQ     .+4
004146  000000                                  HLT
                                        ;TEST THAT MFPD/I PUSHES DESTINATION MEMORY DATA ONTO THE APPROPRIATE
                                        ;STACK,
                                        ;KERNEL MODE,MFPD
004150  010701                          T52:    SCOPE
004152  005037  177776                          CLR     @#PSW           ;KERNEL MODE!!!
004156  012700  001002                          MOV     #TEMP,R0        ;PRESET R0
004162  052737  000000  177776                  BIS     #REG,@#PSW      ;SELECT R0-R5
004170  012700  001004                          MOV     #TEMP+2,R0      ;PRESET R0
004174  012767  177777  174600                  MOV     #-1,TEMP
004202  005067  174576                          CLR     TEMP+2
004206  012706  000500                          MOV     #KPTR,KSP       ;SET KERNEL STACK POINTER
004212  012716  125252                          MOV     #125252,(KSP)   ;PRESET KERNEL STACK
004216  006520                                  MFPI    (R0)+           ;-(KSP)-(R0)+,R0-TEMP+2,TEMP+2-0

004220  013702  177776                          MOV     @#PSW,R2
004224  022702  000004                          CMP     #REG+E,R2
004230  001401                                  BEQ     .+4
004232  000000                                  HLT
004234  022706  000476                          CMP     #KPTR-2,KSP
004240  001401                                  BEQ     .+4
004242  000000                                  HLT
004244  005716                                  TST     (KSP)
004246  001401                                  BEQ     .+4
004250  000000                                  HLT

                                        ;USER MODE MFPI
004252  010701                          T54:    SCOPE
004254  012737  140000  177776                  MOV     #UM,@#PSW
004262  012703  001004                          MOV     #TEMP+2,R3
004266  052737  000340  177776                  BIS     #REG+PRTY7,@#PSW
004274  012703  001006                          MOV     #TEMP+4,R3
004300  005067  174476                          CLR     TEMP
```

```
004304  012767  177777  174472          MOV     #-1,TEMP+2
004312  012706  000700                  MOV     #UPTR,USP
004316  012716  125252                  MOV     #125252,(USP)
004322  006563  177776                  MFPI    -2(R3)          ;-(USP)<--2(R3),R3=#TEMP+4,TEMP+2=-1

004326  013700  177776                  MOV     @#PSW,R0
004332  010602                          MOV     USP,R2
004334  042737  140000  177776          BIC     #UM,@#PSW
004342  022700  140350                  CMP     #UM+PRTY7+N,R0
004346  001401                          BEQ     .+4
004350  000000                          HLT
004352  022702  000676                  CMP     #UPTR-2,R2
004356  001401                          BEQ     .+4
004360  000000                          HLT
004362  005112                          COM     (R2)
004364  001401                          BEQ     .+4
004366  000000                          HLT
                                ;TEST OVERFLOW (YELLOW) USING MFPD INSTRUCTION
004370  010701                  T55:    SCOPE
004372  012737  030000  177776          MOV     #PUM,@#PSW      ;KERNEL MODE!!!,PREV USER MODE!!
004400  012706  001000                  MOV     #YELPTR,KSP     ;SET STACK PTR AT TOP OF YELLOW ZONE
004404  012767  177777  174370          MOV     #-1,TEMP        ;PRESET DATA
004412  005066  177776                  CLR     -2(KSP)         ;PRESET STACK DATA
004416  012737  004444  000004          MOV     #T55A,@#ERRVEC  ;LOAD ERROR TRAP VECTOR
004424  005037  000006                  CLR     @#ERRVEC+2
004430  012737  000400  177774          MOV     #400,@#SLR      ;SET STACK LIMIT =1000
004436  006567  174340                  MFPI    TEMP            ;PUSH TEMP ONTO KERNEL STACK
                                                                ;SHOULD OVERFLOW STACK
004442  000000                  T55AA:  HLT                     ;ERROR! FAILED TO TRAP ON OVERFLOW
004444  022767  177777  174324  T55A:   CMP     #-1,YELPTR-2    ;CHECK THAT MFPD PUSHED DATA
004452  001401                          BEQ     .+4             ;ONTO STACK
004454  000000                          HLT                     ;ERROR! MFPD FAILED TO PUSH DATA
004456  022767  030010  174310          CMP     #PUM+N,YELPTR-4 ;CHECK SAVED STATUS ON TRAP
004464  001401                          BEQ     .+4
004466  000000                          HLT                     ;ERROR! INCORRECT STATUS SAVED
004470  022767  004442  174274          CMP     #T55AA,YELPTR-6 ;CHECK SAVED PC ON STACK
004476  001401                          BEQ     .+4
004500  000000                          HLT                     ;ERROR! INCORRECCT PC SAVED ON STACK
004502  005037  177774                  CLR     @#SLR           ;CLEAR STACK LIMIT REGISTER

                                ;TEST OVERFLOW (RED) USING MFPI INSTRUCTION
004506  010701                  T56:    SCOPE
004510  012737  004562  000004          MOV     #T56A,@#ERRVEC  ;SET ERROR TRAP VECTOR
004516  012737  030340  177776          MOV     #PUM+PRTY7,@#PSW        ;KERNEL MODE!!!, PREV USER MODE!!
004524  012706  000736                  MOV     #REDPTR,KSP     ;SET STACK PTR TO TOP OF RED ZONE
004530  012766  177777  177776          MOV     #-1,-2(KSP)     ;PRESET RED LOCATION=-1
004536  005067  174240                  CLR     TEMP            ;(TEMP)WILL BE THE DATA MOVED
                                                                ;TO RED LOCATION
004542  012703  001004                  MOV     #TEMP+2,R3      ;LOAD INDEX REGISTER
004546  012737  000400  177774          MOV     #400,@#SLR      ;SET STACK LIMIT=1000
004554  006563  177776                  MFPI    -2(R3)          ;-(KSP)=TEMP SHOULD OVER
                                                                ;FLOW (RED)
004560  000000                  T56AA:  HLT                     ;ERROR! FAILED TO TRAP ON 'RED'
                                                                ;OVERFLOW
```

```
004562  022737  177777  000734  T56A1:  CMP     #-1,@#REDPTR,2  ;TEST THAT MFPI DID NOT WRITE
004570  001401                          BEQ     .+4             ;INTO 'RED' LOCATION
004572  000000                          HLT                     ;ERROR!
004574  005706                          TST     KSP             ;STACK SHOULD HAVE GONE TO 0
004576  001401                          BEQ     .+4
004600  000000                          HLT
004602  022737  030344  000002          CMP     #PUM+PRTY7+2,@#2 ;OLD STATUS SHOULD BE IN 2
004610  001401                          BEQ     .+4
004612  000000                          HLT                     ;ERROR!
004614  022737  004560  000000          CMP     #T56AA,@#0      ;AND RETURN IN 0
004622  001401                          BEQ     .+4
004624  000000                          HLT                     ;ERROR! INCORRECT PC IN 0
004626  005037  177774                  CLR     @#SLR
004632  012737  000006  000004          MOV     #ERRVEC+2,@#ERRVEC;RESTORE ERROR VECTOR

                                ;TEST TRAP & RETURN USER-KERNEL-USER
004640  010701                  T59:    SCOPE
004642  012706  000500                  MOV     #KPTR,KSP       ;SET KERNEL STACK POINTER
004646  012737  000340  000036          MOV     #PRTY7,@#TRAPVEC+2
004654  012737  004744  000034          MOV     #T57A,@#TRAPVEC
004662  012737  140000  177776          MOV     #UM,@#PSW       ;USER MODE!!!
004670  005002                          CLR     R2
004672  104400                          TRAP                    ;TRAP & ENTER KERNEL MODE
004674  013767  177776  174100  T59AA:  MOV     @#PSW,TEMP
004702  042737  140000  177776          BIC     #UM,@#PSW               ;KERNEL MODE!!!
004710  022767  004674  173556          CMP     #T57AA,KPTR+4   ;CHECK THAT RETURN ADDRESS IS ON
004716  001401                          BEQ     .+4             ;KERNEL STACK
004720  000000                          HLT                     ;ERROR!RETURN ADDRESS NOT ON STACK
004722  022767  140004  174052          CMP     #UM+2,TEMP      ;CHECK THAT CORRECT @#PSW WAS
004730  001401                          BEQ     .+4             ;RESTORED ON THE RETURN
004732  000000                          HLT                     ;ERROR! INCORRECT STATUS WAS RETURNED
                                                                ;BY KERNEL FROM TRAP
004734  005102                          COM     R2              ;CHECK THAT TRAP ROUTINE WAS EXECUTED
004736  001401                          BEQ     .+4
004740  000000                          HLT                     ;ERROR! KERNEL DID NOT DO COM R2
                                                                ;(AT T57A)
004742  000402                          BR      T57EX           ;EXIT TEST
004744  005102                  T59A:   COM     R2              ;COMPLEMENT R2
004746  000002                          RTI                     ;AND EXIT
004750  000240                  T59EX:  NOP

                                ;TEST THAT MFPD/I CAN PUSH ONTO CURRENT STACK (AS DETERMINED BY PS15 &
                                ;PS14) THE PREVIOUS MODES STACK POINTER (AS DETERMINED BY PS13 &PS12)
                                ;-(KSP)-KSP,MFPD
004752  010701                  T60:    SCOPE
004754  005037  177776                  CLR     @#PSW           ;KERNEL MODE!!!, PREV KERNEL MODE!!
004760  012706  000500                  MOV     #KPTR,KSP       ;SET KERNEL STACK POINTER
004764  006506                          MFPI    KSP             ;-(KSP)-KSP
004766  022767  000500  173502          CMP     #KPTR,KPTR+2    ;TEST THAT VALUE OF KERNEL STACK POINTER
004774  001401                          BEQ     .+4             ;WAS PUSHED ONTO KERNEL STACK
004776  000000                          HLT                     ;ERROR!


                                ;-(KSP)-USP,MFPD
005000  010701                  T62:    SCOPE
```

```
005002  012737  030000  177776          MOV     #KM+PUM,@#PSW   ;KERNEL MODE!!!, PREV USER MODE!!
005010  012706  000500                  MOV     #KPTR,KSP       ;SET KERNEL STACK POINTER
005014  012716  177777                  MOV     #-1,(KSP)
005020  006606                          MTPI    USP             ;SET USER STACK POINTER USP=(KSP)+
005022  005166  177776                  COM     -2(KSP)         ;PRESET KERNEL STACK
005026  006506                          MFPI    USP             ;-(KSP)=USP
005030  022716  177777                  CMP     #-1,(KSP)       ;CHECK THAT USER STACK POINTER WAS
005034  001401                          BEQ     .+4             ;PUSHED ONTO KERNEL STACK
005036  000000                          HLT                     ;ERROR!

                                ;=(USP)+USP,MFPD
005040  010701                  T65:    SCOPE
005042  012737  030000  177776          MOV     #PUM,@#PSW      ;KERNEL MODE!!!, PREV USER MODE!!
005050  012706  000500                  MOV     #KPTR,KSP       ;SET KERNEL STACK POINTER
005054  012716  000700                  MOV     #UPTR,(KSP)
005060  006606                          MTPI    USP             ;SET USER STACK POINTER
005062  005067  173610                  CLR     UPTR-2
005066  052737  140000  177776          BIS     #UM,@#PSW               ;USER MODE!!!, PREV USER MODE!!!
005074  006506                          MFPI    USP             ;PUSH USER STACK POINTER ONTO USER STACK
005076  042737  140000  177776          BIC     #UM,@#PSW               ;KERNEL MODE!!!, PREV USER MODE!!
005104  006506                          MFPI    USP             ;PUSH USER STACK POINTER ONTO KERNEL STACK
005106  022716  000676                  CMP     #UPTR-2,(KSP)   ;CHECK THAT USER STACK POITER WAS
005112  001401                          BEQ     .+4             ;PUSHED PROPERLY (ONCE)
005114  000000                          HLT                     ;ERROR!
005116  022767  000700  173552          CMP     #UPTR,UPTR-2    ;CHECK THAT USER STACK POINTER IS ON THE
005124  001401                          BEQ     .+4             ;USERS STACK
005126  000000                          HLT                     ;ERROR!

                                ;=(KSP)+KSP,MFPI
005130  010701                  T66:    SCOPE
005132  005037  177776                  CLR     @#PSW           ;KERNEL MODE!!!, PREV KERNEL MODE!!
005136  012706  000500                  MOV     #KPTR,KSP       ;SET KERNEL STACK POINTER
005142  006506                          MFPI    KSP             ;PUSH KERNEL STACK POINTER ONTO KERNEL
                                                                ;STACK
005144  022767  000500  173324          CMP     #KPTR,KPTR-2    ;CHECK RESULT
005152  001401                          BEQ     .+4
005154  000000                          HLT                     ;ERROR!

                                ;=(KSP)+USP,MFPI
005156  010701                  T70:    SCOPE
005160  012737  030000  177776          MOV     #PUM,@#PSW      ;KERNEL MODE!!!, PREV USER MODE!!
005166  012706  000500                  MOV     #KPTR,KSP       ;SET KERNEL STACK POINTER
005172  012716  177777                  MOV     #-1,(KSP)
005176  006606                          MTPI    USP             ;SET USER STACK POINTER
005200  005166  177776                  COM     -2(KSP)         ;PRESET KERNEL STACK
005204  006506                          MFPI    USP             ;PUSH USER STACK POINTER ONTO KERNEL STACK
005206  022716  177777                  CMP     #-1,(KSP)       ;CHECK RESULT
005212  001401                          BEQ     .+4
005214  000000                          HLT                     ;ERROR! USER STACK POINTER NOT ON KERNEL STACK

                                ;=(USP)+USP,MFPI
005216  010701                  T73:    SCOPE
005220  012737  030000  177776          MOV     #PUM,@#PSW      ;KERNEL MODE!!!, PREV USER MODE!!
005226  012706  000500                  MOV     #KPTR,KSP               ;SET KERNEL STACK POINTER
```

```
005232  012716  000700                  MOV     #UPTR,(KSP)
005236  006606                          MTPI    USP             ;SET USER STACK POINTER
005240  005067  173432                  CLR     UPTR-2          ;PRESET USER STACK
005244  052737  140000  177776          BIS     #UM,@#PSW               ;USER MODE!!!, PREV USER MODE!!
005252  006506                          MFPI    USP             ;-(USP)-USP
005254  042737  140000  177776          BIC     #UM,@#PSW               ;KERNEL MODE!!!
005262  006506                          MFPI    USP             ;GET USER STACK POINTER
005264  022716  000676                  CMP     #UPTR-2,(KSP)   ;CHECK THAT USER STACK POINTER WAS
005270  001401                          BEQ     .+4             ;PUSHED ONCE
005272  000000                          HLT                     ;ERROR!
005274  022767  000700  173374          CMP     #UPTR,UPTR-2    ;CHECK THAT USER STACK POINTER WAS PUSHED
005302  001401                          BEQ     .+4             ;ONTO USER STACK
005304  000000                          HLT                     ;ERROR!

                                ;TEST THAT ILLEGAL MODE DOES NOT HANG BUS;
005306  010701                  T741:   SCOPE
005310  012737  100000  177776          MOV     #IM,@#PSW               ;ILLEGAL MODE!!!
005316  013700  177776                  MOV     @#PSW,R0                ;GET ILLEGAL MODE
005322  005037  177776                  CLR     @#PSW           ;KERNEL MODE!!
005326  022700  100000                  CMP     #IM,R0          ;CHECK THAT ILLEGAL MODE WAS SET
005332  001401                          BEQ     .+4             ;INTO STATUS
005334  000000                          HLT

                                ;TEST THAT ILLEGAL MODE DOES NOT HANG BUS;
005336  010701                  T751:   SCOPE
005340  012737  040000  177776          MOV     #IM1,@#PSW              ;ILLEGAL MODE!!!
005346  013700  177776                  MOV     @#PSW,R0                ;GET ILLEGAL MODE
005352  005037  177776                  CLR     @#PSW           ;KERNEL MODE!!
005356  022700  040000                  CMP     #IM1,R0         ;CHECK THAT ILLEGAL MODE WAS SET
005362  001401                          BEQ     .+4             ;INTO STATUS
005364  000000                          HLT

                                ;TEST THAT KERNEL CAN GET DATA FROM USER STACK
005366  010701                  T761:   SCOPE
005370  012737  030000  177776          MOV     #KM+PUM,@#PSW   ;KERNEL MODE!!!, PREV USER MODE!!
005376  012706  000500                  MOV     #KPTR,KSP       ;SET KERNEL STACK POINTER
005402  012716  000700                  MOV     #UPTR,(KSP)
005406  006606                          MTPI    USP             ;SET USER STACK POINTER
005410  005067  173264                  CLR     UPTR            ;PRESET USER STACK
005414  005016                          CLR     (KSP)           ;PRESET KERNEL STACK
005416  012766  177777  177776          MOV     #-1,-2(KSP)
005424  006506                          MFPI    USP             ;-(KSP)-USP
005426  006576  000000                  MFPI    @(KSP)          ;LIKE MOV  @(6),-(6)
005432  000240                          NOP
005434  013703  177776                  MOV     @#PSW,R3                ;SAVE STATUS RESULT
005440  022767  000700  173032          CMP     #UPTR,KPTR      ;CHECK THAT USER STACK POINTER WAS
005446  001401                          BEQ     .+4             ;PUSHED ONTO KERNEL STACK
005450  000000                          HLT                     ;ERROR!
005452  022706  000476                  CMP     #KPTR-2,KSP     ;CHECK THAT KERNEL STACK POINTER IS POS-
005456  001401                          BEQ     .+4             ;ITIONED PROPERLY
005460  000000                          HLT                     ;ERROR! INCORRECT KERNEL STACK POINTER
005462  005716                          TST     (KSP)           ;CHECK THAT CORRECT DATA
005464  001401                          BEQ     .+4             ;WAS PUSHED ONTO KERNEL STACK
005466  000000                          HLT                     ;ERROR!
```

```
005470  022703  030004                CMP     #KM+PUM+Z,R3    ;CHECK STATUS
005474  001401                        BEQ     .+4
005476  000000                        HLT                     ;ERROR: INCORRECT STATUS

                                ;CHECK THAT MTPD CAN LOAD MEMORY ADDRESS DM=7,PC
005500  010701                T102:   SCOPE
005502  012737  030000  177776        MOV     #KM+PUM,@#PSW   ;KERNEL MODE!!!, PREV USER MODE!!
005510  012706  000500                MOV     #KPTR,KSP       ;SET KERNEL STACK PTR
005514  005016                        CLR     (KSP)           ;PUT DATA ON STACK
005516  012737  001002  001004        MOV     #TEMP,@#TEMP+2  ;LOAD ADDRESS
005524  012767  177777  173250        MOV     #-1,TEMP        ;PRESET DATA
005532  000277                        SCC                     ;PRESET CC'S
005534  006677  173244                MTPI    @TEMP+2         ;TEMP+(KSP)+
005540  013703  177776                MOV     @#PSW,R3                ;CHECK CC'S
005544  022703  030005                CMP     #PUM+Z+C,R3     ;CHECK CC'S
005550  001401                        BEQ     .+4
005552  000000                        HLT                     ;ERROR: INCORRECT CC'S AFTER MTPD
005554  005737  001002                TST     @#TEMP          ;CHECK RESULT
005560  001401                        BEQ     .+4
005562  000000                        HLT                     ;ERROR: INCORRECT RESULT

                                ;CHECK THAT MTPI CAN LOAD MEMORY ADDRESS DM=7
005564  010701                T103:   SCOPE
005566  012737  030000  177776        MOV     #KM+PUM,@#PSW   ;KERNEL MODE!!!
005574  012706  000500                MOV     #KPTR,KSP       ;SET KERNEL STACK PTR
005600  012716  177777                MOV     #-1,(KSP)       ;LOAD DATA ONTO STACK
005604  012704  177776                MOV     #-2,R4          ;LOAD INDEX REGISTER
005610  005067  173166                CLR     TEMP            ;PRESET DATA
005614  012767  001002  173162        MOV     #TEMP,TEMP+2
005622  006674  001006                MTPI    @TEMP+4(R4)     ;TEMP+(KSP)+
005626  013703  177776                MOV     @#PSW,R3                ;SAVE STATUS RESULT
005632  022706  000502                CMP     #KPTR+2,KSP     ;CHECK THAT KSP POPPED
005636  001401                        BEQ     .+4
005640  000000                        HLT                     ;ERROR: INCORRECT STACK PTR
005642  022703  030010                CMP     #PUM+N,R3       ;CHECK STATUS RESULT
005646  001401                        BEQ     .+4
005650  000000                        HLT                     ;ERROR: INCORRECT STATUS
005652  005267  173124                INC     TEMP            ;CHECK RESULT
005656  001401                        BEQ     .+4
005660  000000                        HLT                     ;ERROR: INCORRECT RESULT

                                ;TEST THAT MTPD/I CAN LOAD PC
005662  010701                T104:   SCOPE
005664  012737  000000  177776        MOV     #KM,@#PSW       ;KERNEL MODE!!!
005672  012706  000500                MOV     #KPTR,KSP       ;SET KERNEL STACK PTR
005676  012716  005710                MOV     #T104A,(KSP)    ;PUT NEW PC ON STACK
005702  000277                        SCC                     ;PRESET CC'S
005704  006607                        MTPI    PC              ;PC+(KSP)+
005706  000000                        HLT                     ;ERROR: MTPD FAILED TO SET PC
005710  100001                T104A:  BPL     .+4
005712  000000                        HLT                     ;ERROR: 'N' FAILED TO CLEAR IN STATUS
005714  103401                        BCS     .+4
005716  000000                        HLT                     ;ERROR: 'C' WAS CLEARED BY MTPD
```

```
                                        ;USER MODE
005720  010701                          T106:   SCOPE
005722  012737  170000  177776                  MOV     #UM+PUM,@#PSW   ;USER MODE!!!
005730  012706  000700                          MOV     #UPTR,USP       ;SET USER STACK PTR
005734  012716  005752                          MOV     #T106A,(USP)    ;PUT NEW PC ON STACK
005740  000277                                  SCC                     ;PRESET CC'S
005742  006607                                  MTPI    PC              ;PC=(USP)+
005744  005037  177776                          CLR     @#PSW           ;KERNEL MODE!!!
005750  000000                                  HLT                     ;ERROR! MTPD FAILED TOMLOAD PC
005752  013705  177776          T106A:  MOV     @#PSW,R5                        ;SAVE STATUS
005756  005037  177776                          CLR     @#PSW           ;KERNEL MODE!!!
005762  022705  170001                          CMP     #UM+PUM+C,R5    ;CHECK STATUS
005766  001401                                  BEQ     .+4
005770  000000                                  HLT

                                        ;TEST ERROR TRAP (ODD ADDRESS) MFPD/I
005772  010701                          T107:   SCOPE
005774  005037  177776                          CLR     @#PSW           ;KERNEL MODE!!!
006000  012706  000500                          MOV     #KPTR,KSP       ;SET KERNEL STACK PTR
006004  012737  006022  000004                  MOV     #T107A,@#ERRVEC ;LOAD ERROR VECTOR
006012  000277                                  SCC                     ;PRESET CC'S
006014  006567  171761                          MFPI    1               ;ODD ADDRESS SHOULD TRAP
006020  000000                          T107AA: HLT                     ;ERROR! FAILED TO TRAP ON ODD ADDRESS
006022  022706  000474                  T107A:  CMP     #KPTR+4,KSP     ;CHECK THAT STACK PTR WAS PUSHED
006026  001401                                  BEQ     .+4             ;PROPERLY (2 PUSHES)
006030  000000                                  HLT                     ;ERROR! INCORRECT STACK PTR AFTER ERROR
006032  022726  006020                          CMP     #T107AA,(KSP)+  ;CHECK RETURN PC ON STACK
006036  001401                                  BEQ     .+4
006040  000000                                  HLT                     ;ERROR! RETURN PC NOT ON STACK
006042  022716  000017                          CMP     #17,(KSP)       ;CHECK SAVED STATUS ON STACK
006046  001401                                  BEQ     .+4
006050  000000                                  HLT                     ;ERROR! INCORRECT STATUS SAVED ON STACK
                                        ;USER MODE, TIME OUT
006052  010701                          T110:   SCOPE
006054  012737  140000  177776                  MOV     #UM,@#PSW               ;USER MODE!!!
006062  012706  000700                          MOV     #UPTR,USP       ;SET USER STACK
006066  012737  140000  000006                  MOV     #UM,@#ERRVEC+2  ;LOAD 'NEW' STATUS
006074  012737  006114  000004                  MOV     #T110A,@#ERRVEC ;AND PC
006102  006537  177702                          MFPI    @#177702        ;177702 IS NON-EXISTANT ADRS
006106  005037  177776          T110AA: CLR     @#PSW           ;KERNEL MODE!!!
006112  000000                                  HLT                     ;ERROR! DID NOT TRAP ON NON ADRS
006114  010603                          T110A:  MOV     USP,R3          ;SAVE USER STACK PTR
006116  042737  140000  177776                  BIC     #UM,@#PSW               ;KERNEL MODE!!!
006124  022703  000674                          CMP     #UPTR+4,R3      ;CHECK USER STACK PTR
006130  001401                                  BEQ     .+4
006132  000000                                  HLT                     ;ERROR! INCORRECT USP AFTER ERROR TRAP
006134  022723  006106                          CMP     #T110AA,(R3)+   ;CHECK RETURN PC ON USER STACK
006140  001401                                  BEQ     .+4
006142  000000                                  HLT                     ;ERROR! RETURN PC NOT ON USER STACK
006144  022713  140000                          CMP     #UM,(R3)        ;CHECK SAVED STATUS
006150  001401                                  BEQ     .+4
006152  000000                                  HLT                     ;ERROR! INCORRECT STATUS SAVED ON STACK

                                        ;USER MODE,ODD ADDRESS
```

```
006154  010701                  T111:   SCOPE
006156  012737  140000  177776          MOV     #UM,@#PSW       ;USER MODE!!!
006164  012706  000700                  MOV     #UPTR,USP       ;SET USER STACK PTR
006170  012737  006216  000004          MOV     #T111A,@#ERRVEC ;LOAD ERROR TRAP VECTOR
006176  012737  140000  000006          MOV     #UM,@#ERRVEC+2
006204  006567  171567                  MFPI    -1              ;ODD ADDRESS SHOULD TRAP
006210  005037  177776          T111AA: CLR     @#PSW           ;KERNEL MODE!!!
006214  000000                          HLT                     ;ERROR: FAILED TO TRAP
006216  010603                  T111A:  MOV     USP,R3          ;SAVE USER STACK PTR
006220  042737  140000  177776          BIC     #UM,@#PSW               ;KERNEL MODE!!!
006226  022703  000674                  CMP     #UPTR-4,R3      ;CHECK USER STACK PTR
006232  001401                          BEQ     .+4
006234  000000                          HLT                     ;ERROR: INCORRECT USER STACK POINTER
006236  022713  006210                  CMP     #T111AA,(R3)    ;CHECK RETURN SDDRESS ON USER STACK
006242  001401                          BEQ     .+4
006244  000000                          HLT                     ;ERROR: RETURN PC NOT ON USER STACK
006246  012737  000006  000004          MOV     #ERRVEC+2,@#ERRVEC;RESTORE ERROR TRAP TO HALT
006254  005067  171526                  CLR     ERRVEC+2

                                ;TEST THAT MTPD INSTRUCTION CAN LOAD DATA TO AN ADDRESS VIA THE STACK
                                ;KERNEL MODE,PREVIOUS USER MODE
006260  010701                  T112:   SCOPE
006262  012737  030000  177776          MOV     #KM+PUM,@#PSW   ;KERNEL MODE!!!, PREV USER MODE!!
006270  012706  000500                  MOV     #KPTR,KSP       ;SET KERNEL STACK PTR
006274  012746  000700                  MOV     #UPTR,-(KSP)
006300  006606                          MTPI    USP             ;SET USER STACK PTR
006302  012746  001002                  MOV     #TEMP,-(KSP)    ;PUT ADDRESS ON THE STACK
006306  012746  177777                  MOV     #-1,-(KSP)      ;PUT DATA ON THE STAK
006312  005037  001002                  CLR     @#TEMP          ;PRESET DATA
006316  006636                          MTPI    @(KSP)+         ;MOVE #-1 TO TEMP
006320  022706  000500                  CMP     #KPTR,KSP       ;CHECK STACK PTR AFTER MTPD
006324  001401                          BEQ     .+4
006326  000000                          HLT                     ;ERROR: INCORRECT STACK PTR AFTER MTPD
006330  005267  172446                  INC     TEMP            ;CHECK THAT DATA WAS MOVED TO TEMP
006334  001401                          BEQ     .+4
006336  000000                          HLT                     ;ERROR: DATA NOT IN TEMP
006340  006506                          MFPI    USP             ;GET USER STACK PTR
006342  022716  000700                  CMP     #UPTR,(KSP)     ;CHECK THAT USER STACK PTR NOT CHANGED
006346  001401                          BEQ     .+4             ;BY MTPD INSTRUCTION
006350  000000                          HLT                     ;ERROR: USP WAS CHANGED BY MTPD INST;

006352  005267  172422          END:    INC     ICNT            ;INCREMENT PASS COUNT
006356  026727  172416  000144          CMP     ICNT,#100.      ;100 PASSES COMPLETED?
006364  001402                          BEQ     DONE
006366  000167  172440                  JMP     BEGIN
006372  032767  010000  171170  DONE:   BIT     #10000,SWR              ;INHIBIT BELL?
006400  001401                          BEQ     .+4
006402  000414                          BR      LOGICT
006404  012767  000007  171154          MOV     #7,TPB          ;RING BELL
006412  105767  171146                  TSTB    TPS
006416  100375                          BPL     .-4
006420  012767  000177  171140          MOV     #177,TPB
006426  105767  171132                  TSTB    TPS
006432  100375                          BPL     .-4
```

```
006434  013701  000042          LOGIC7: MOV     @#42,R1         ;RETURN TO MONITOR?
006440  001435                          BEQ     LOGICE
006442  000005                          RESET
006444  004711                  LOGIC:  JSR     7,(1)           ;RETURN;
006446  000240                          NOP
006450  000240                          NOP
006452  000240                          NOP
006454  000167  172332          LOGICE: JMP     START
        000001                          .END
```

```
BEGIN    001032          BIT13  = 020000          BIT14  = 040000          BIT15  = 100000
BIT6   = 000100          C      = 000001          DONE     006372          EMTVEC = 000030
END      006352          ERRVEC = 000004          HLT    = 000000          ICNT     001000
IM     = 100000          IM1    = 040000          IOTVEC = 000020          KM     = 000000
KPTR   = 000500          KSP    =X000006          LOGIC    006444          LOGICE   006454
LOGICT   006434          N      = 000010          PC     =X000007          PKM    = 000000
PRTY3  = 000140          PRTY4  = 000200          PRTY7  = 000340          PSW    = 177776
PUM    = 030000          PWRUP    001022          REDPTR = 000736          REG    = 000000
RESVEC = 000010          R0     =X000000          R1     =X000001          R2     =X000002
R3     =X000003          R4     =X000004          R5     =X000005          SCOPE  = 010701
SLR    = 177774          START    001012          SWR    = 177570          TBIT   = 000020
TBITVE = 000014          TEMP     001002          TKB    = 177562          TKS    = 177560
TPB    = 177566          TPS    = 177564          TPVEC  = 000064          TRAPVE = 000034
TRT    = 000003          TRTVEC = 000014          T1       001036          T102     005500
T103     005564          T104     005662          T104A    005710          T106     005720
T106A    005752          T107     005772          T107A    006022          T107AA   006020
T110     006052          T110A    006114          T110AA   006106          T111     006154
T111A    006216          T111AA   006210          T112     006260          T12      001352
T12A     001410          T12AA    001402          T13      001442          T13A     001474
T13AA    001466          T15      001640          T15A     001720          T15AA    001712
T17      002002          T17A     002036          T17B     002050          T17C     002078
T17D     002126          T17E     002150          T17ERR   002250          T17F     002170
T17G     002222          T17X     002256          T18      001534          T19      001576
T21      002274          T22      002356          T25      002440          T26      002530
T30      002620          T31      002714          T31C     002740          T32A     003012
T35      003062          T36      003122          T36A     003160          T36AA    003156
T40      003202          T40A     003252          T40AA    003244          T41      003326
T41A     003374          T41B     003414          T41BB    003444          T43      003472
T43A     003544          T44      003612          T45      003666          T5       001078
T5A      001134          T5AA     001126          T50      003752          T51      004052
T52      004150          T54      004252          T55      004370          T55A     004444
T55AA    004442          T56      004506          T56A     004562          T56AA    004560
T57      004640          T57A     004744          T57AA    004674          T57EX    004750
T60      004752          T62      005000          T65      005040          T66      005130
T7       001242          T7A      001302          T7AA     001274          T70      005154
T73      005216          T74      005306          T75      005336          T76      005364
UM     = 140000          UPTR   = 000700          USP    =X000006          V      = 000002
YELPTR = 001000          Z      = 000004          .      = 006400
```

ERRORS DETECTED:  0

```
*STATV3,STATV3=STATV3;SRC
RUN-TIME:  4  9  0 SECONDS
CORE USED:  3K
```

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DCKBF-A1-D

PRODUCT NAME: STACK LIMIT TEST
STANNDARD ON 11/45
KJ11-A OPTION ON 11/40

DATE CREATED: 15 MAR 1972

MAINTAINER: DIAGNOSTIC GROUP

AUTHORS: JOHN ADAMS

COPYRIGHT(c) 1972
DIGITAL EQUIPMENT CORPORATION
MAYNARD, MASS

1,0 ABSTRACT

THIS PROGRAM INCREMENTLY TESTS THE STACK LIMIT FUNCTION

2,0 REQUIREMENTS

2,1 EQUIPMENT
BASIC 11/45 SYSTEM OR
11/40 WITH STACK LIMIT OPTION

2,2 STORAGE
THIS PROGRAM USES 0 THRU 17500

2,3 PRELIMINARY PROGRAMS
D0AA THRU D0MA

3,0 LOADING PROCEDURE
LOAD PROGRAM USING ABS LOADER

4,0 STARTING PROCEDURE
LOAD ADDRESS 200, PRESS START, THE PROGRAM WILL LOOP
AND RING BELL ON PASS COMPLETION,

5,0 OPERATING PROCEDURE

5,1 SWITCH SETTINGS
NONE

5,2 SUBROUTINE ABSTRACTS

5,2,1 SCOPE
SCOPE IS A MOVE PC,R1 AND STORES THE PC+2 IN R1,

5,2,2 HLT
HLT IS A HALT INSTRUCTION,

6,0 ERRORS
ALL ERRORS WILL CAUSE A HALT
TRAP AND INTERRUPT ERRORS WILL CAUSE A HALT AT VECTOR+2,

6,1 ERROR RECOVERY
PRESS CONTINUE TO PROCEDE TO NEXT TEST

6,2 ERROR LOOPING
TO LOOP ON AN ERROR, PLACE A BRANCH TO THE PREVIOUS SCOPE
INSTRUCTION IN PLACE OF THE HALT INSTRUCTION,
NOTE THAT IF THE ERROR IS INTERMITTANT THAT THE TEST WILL
DROP THRU THE HALT AND PROCEED TO THE NEXT TEST,
THEREFORE, TO LOOP THE TEST CONTINUOSLY REPLACE THE BEQ ,+4
INSTRUCTION IMMEDIATLY PRECEEDING THE HALT WITH A BRANCH
BACK TO THE PREVIOUS SCOPE,

TO LOOP ON TRAP FAILURES, PATCH IN THE FOLLOWING ROUTINE
AT THE ADDRESS OF THE TRAP VECTOR,

```
TRAPVEC:        TRAPVEC+4
TRAPVEC+2:      0
TRAPVEC+4:      012716  ;MOVE SCOPE ADDRESS TO STACK
TRAPVEC+6:      ADDRESS ;ADDRESS OF PREVIOUS SCOPE
TRAPVEC+10:     000006  ;RETURN TO TEST AT SCOPE
```

RESTORE ALL LOCATIONS BEFORE PROCEEDING TO NEXT TEST,

7,0 RESTRICTIONS
NONE

8,0 MISCELLANEOUS
ON TRAP ERRORS THE STACK POINTER(R6) WILL CONTAIN
ADDRESS WHERE THE TRAP OCCURED.

8,1 EXECUTION TIME
EACH PROGRAM TAKES ABOUT 1 MINUTE.

9,0 PROGRAM DESCRIPTIONS

THIS IS A TEST OF THE STACK LIMIT REGISTER AND INSURES
CORRECT OPERATION OF THE RED AND YELLOW ZONE BOUNDARIES.
OVERFLOW TRAPS ARE TESTED FOR ALL VALUES OF THE STACK LIMIT
REGISTER.

```
                                        .TITLE  MAINDEC-11-DCKBF-A PDP11/25-45 STK LIM TEST
                                        .NLIST  MC,MD,SEQ
                                        .LIST   ME
                                        .ABS
                                ;TEST DCKBFA- STACK LIMIT REGISTER TEST
                                ;THE STACK LIMIT REGISTER ALLOWS THE 'OVERFLOW' BOUNDARIES TO BE CHANGED.
                                ;FOR EXAMPLE IF THE STACK LIMIT REGISTER IS CLEAR THE BOUNDARY IS AT
                                ;400 (YELLOW ZONE) AND 340 (RED ZONE). IN ALL CASES THE YELLOW ZONE BOUNDARY
                                ;IS AT 400(8) PLUS THE VALUE IN THE STACK LIMIT REGISTER, AND THE RED
                                ;ZONE BEGINS 20(8) WORDS BELOW THE YELLOW ZONE.
                                ;THIS TEST CHECKS THAT THE STACK LIMIT IS 400 GREATER THAN THE CONTENTS
                                ;OF THE STACK LIMIT REGISTER (CORE PERMITTING), AND CHECKS THE LENGTH
                                ;OF THE YELLOW ZONE AND THE BEGINNING OF THE RED ZONE.


                                ;STARTING PROCEEDURE
                                ;       LOAD ADDRESS=200
                                ;       PRESS START
                                ;       BELL WILL RING WHEN TEST IS COMPLETE

                                ;EQUATE STATEMENTS
000000                          R0=%0
000001                          R1=%1
000002                          R2=%2
000003                          R3=%3
000004                          R4=%4
000005                          R5=%5
000006                          R6=%6
000006                          SP=%6
000007                          PC=%7

                                ;REGISTER ADDRESSES
177776                                  PSW=177776              ;ADDRESS OF PROCESSER STATUS WORD
177770                                  UBREAK=177770           ;ADDRESS OF PDP11/45 MICRO BREAK REGISTER
177570                                  SWR=177570              ;ADDRESS OF CONSOLE SWITCH REGISTER
177564                                  TPS=177564
177566                                  TPB=177566
177570                                  DISPLAY=177570          ;ADDRESS OF CONSOLE DISPLAY REGISTER
000000                          HLT=HALT
022626                          POP2=22626
010701                          SCOPE=010701                    ;MOVE PC TO R1
000340                          PRTY7=340                       ;PRIORITY LEVEL 7
000004                          ERRVEC=4                        ;ADDRESS OF ERROR VECTOR

000000  000002                          .+2
000002  000000                          HALT
000004  000006                          .+2
000006  000000                          HALT
000010  000012                          .+2
000012  000000                          HALT
000014  000016                          .+2
000016  000000                          HALT
000020  000022                          .+2
000022  000000                          HALT
```

```
000024  000026                  ,+2
000026  000000                  HALT
000030  000032                  ,+2
000032  000000                  HALT
000034  000036                  ,+2
000036  000000                  HALT
000040  000042                  ,+2
000042  000000                  HALT
000044  000046                  ,+2
000046  000000                  HALT
000050  000052                  ,+2
000052  000000                  HALT
000054  000056                  ,+2
000056  000000                  HALT
000060  000062                  ,+2
000062  000000                  HALT
000064  000066                  ,+2
000066  000000                  HALT
000070  000072                  ,+2
000072  000000                  HALT
000074  000076                  ,+2
000076  000000                  HALT
000100  000102                  ,+2
000102  000000                  HALT
000104  000106                  ,+2
000106  000000                  HALT
000110  000112                  ,+2
000112  000000                  HALT
000114  000116                  ,+2
000116  000000                  HALT
000120  000122                  ,+2
000122  000000                  HALT
000124  000126                  ,+2
000126  000000                  HALT
000130  000132                  ,+2
000132  000000                  HALT
000134  000136                  ,+2
000136  000000                  HALT
000140  000142                  ,+2
000142  000000                  HALT
000144  000146                  ,+2
000146  000000                  HALT
000150  000152                  ,+2
000152  000000                  HALT
000154  000156                  ,+2
000156  000000                  HALT
000160  000162                  ,+2
000162  000000                  HALT
000164  000166                  ,+2
000166  000000                  HALT
000170  000172                  ,+2
000172  000000                  HALT
000174  000176                  ,+2
000176  000000                  HALT
```

```
000200  000202                  .+2
000202  000000                  HALT
000204  000206                  .+2
000206  000000                  HALT
000210  000212                  .+2
000212  000000                  HALT
000214  000216                  .+2
000216  000000                  HALT
000220  000222                  .+2
000222  000000                  HALT
000224  000226                  .+2
000226  000000                  HALT
000230  000232                  .+2
000232  000000                  HALT
000234  000236                  .+2
000236  000000                  HALT
000240  000242                  .+2
000242  000000                  HALT
000244  000246                  .+2
000246  000000                  HALT
000250  000252                  .+2
000252  000000                  HALT
000254  000256                  .+2
000256  000000                  HALT
000260  000262                  .+2
000262  000000                  HALT
000264  000266                  .+2
000266  000000                  HALT
000270  000272                  .+2
000272  000000                  HALT
000274  000276                  .+2
000276  000000                  HALT
000300  000302                  .+2
000302  000000                  HALT
000304  000306                  .+2
000306  000000                  HALT
000310  000312                  .+2
000312  000000                  HALT
000314  000316                  .+2
000316  000000                  HALT
000320  000322                  .+2
000322  000000                  HALT
000324  000326                  .+2
000326  000000                  HALT
000330  000332                  .+2
000332  000000                  HALT
000334  000336                  .+2
000336  000000                  HALT
000340  000342                  .+2
000342  000000                  HALT
000344  000346                  .+2
000346  000000                  HALT
000350  000352                  .+2
000352  000000                  HALT
```

```
000354  000356                  .+2
000356  000000                  HALT
000360  000362                  .+2
000362  000000                  HALT
000364  000366                  .+2
000366  000000                  HALT
000370  000372                  .+2
000372  000000                  HALT
000374  000376                  .+2
000376  000000                  HALT
000400  000402                  .+2
000402  000000                  HALT
000404  000406                  .+2
000406  000000                  HALT
000410  000412                  .+2
000412  000000                  HALT
000414  000416                  .+2
000416  000000                  HALT
000420  000422                  .+2
000422  000000                  HALT
000424  000426                  .+2
000426  000000                  HALT
000430  000432                  .+2
000432  000000                  HALT
000434  000436                  .+2
000436  000000                  HALT
000440  000442                  .+2
000442  000000                  HALT
000444  000446                  .+2
000446  000000                  HALT
000450  000452                  .+2
000452  000000                  HALT
000454  000456                  .+2
000456  000000                  HALT
000460  000462                  .+2
000462  000000                  HALT
000464  000466                  .+2
000466  000000                  HALT
000470  000472                  .+2
000472  000000                  HALT
000474  000476                  .+2
000476  000000                  HALT
000500  000502                  .+2
000502  000000                  HALT
000504  000506                  .+2
000506  000000                  HALT
000510  000512                  .+2
000512  000000                  HALT
000514  000516                  .+2
000516  000000                  HALT
000520  000522                  .+2
000522  000000                  HALT
000524  000526                  .+2
000526  000000                  HALT
```

```
000530  000532                  .+2
000532  000000                  HALT
000534  000536                  .+2
000536  000000                  HALT
000540  000542                  .+2
000542  000000                  HALT
000544  000546                  .+2
000546  000000                  HALT
000550  000552                  .+2
000552  000000                  HALT
000554  000556                  .+2
000556  000000                  HALT
000560  000562                  .+2
000562  000000                  HALT
000564  000566                  .+2
000566  000000                  HALT
000570  000572                  .+2
000572  000000                  HALT
000574  000576                  .+2
000576  000000                  HALT
000600  000602                  .+2
000602  000000                  HALT
000604  000606                  .+2
000606  000000                  HALT
000610  000612                  .+2
000612  000000                  HALT
000614  000616                  .+2
000616  000000                  HALT
000620  000622                  .+2
000622  000000                  HALT
000624  000626                  .+2
000626  000000                  HALT
000630  000632                  .+2
000632  000000                  HALT
000634  000636                  .+2
000636  000000                  HALT
000640  000642                  .+2
000642  000000                  HALT
000644  000646                  .+2
000646  000000                  HALT
000650  000652                  .+2
000652  000000                  HALT
000654  000656                  .+2
000656  000000                  HALT
000660  000662                  .+2
000662  000000                  HALT
000664  000666                  .+2
000666  000000                  HALT
000670  000672                  .+2
000672  000000                  HALT
000674  000676                  .+2
000676  000000                  HALT
000700  000702                  .+2
000702  000000                  HALT
```

```
000704  000706                          ,+2
000706  000000                          HALT
000710  000712                          ,+2
000712  000000                          HALT
000714  000716                          ,+2
000716  000000                          HALT
000720  000722                          ,+2
000722  000000                          HALT
000724  000726                          ,+2
000726  000000                          HALT
000730  000732                          ,+2
000732  000000                          HALT
000734  000736                          ,+2
000736  000000                          HALT
000740  000742                          ,+2
000742  000000                          HALT
000744  000746                          ,+2
000746  000000                          HALT
000750  000752                          ,+2
000752  000000                          HALT
000754  000756                          ,+2
000756  000000                          HALT
000760  000762                          ,+2
000762  000000                          HALT
000764  000766                          ,+2
000766  000000                          HALT
000770  000772                          ,+2
000772  000000                          HALT
000774  000776                          ,+2
000776  000000                          HALT

        000200                          ,=200
000200  000167  000604                  JMP     START

        000500                          ,=500
                        ;TAGS
000500  000000          ICNT:   0               ;CONTAINS PASS COUNT
000502  000760          SPBOT:  760
000504  177774          SLR:    177774          ;ADDRESS OF STACK LIMIT REGISTER
000506  177775          SLH:    177775          ;HIGH (ODD BYTE)
000510  000000          TEMP:   0
```

```
                001010                          .=1010
001010  016706  177466          START:  MOV     SPBOT,%6        ;INITIALIZE STACK POINTER
001014  012737  000340  177776          MOV     #PRTY7,@#PSW    ;LOCK OUT INTERRUPTS
001022  005067  177452                  CLR     ICNT            ;CLEAR PASS COUNT

001026  016706  177450          BEGIN:  MOV     SPBOT,%6        ;INITIALIZE STACK POINTER
001032  016737  177442  177570          MOV     ICNT,@#DISPLAY  ;DISPLAY PASS COUNT
001040  032737  000400  177570          BIT     #400,@#SWR      ;LOAD PDP11/45 MICRO BREAK REGISTER?
001046  001403                          BEQ     .+10
001050  113737  177570  177770          MOVB    @#SWR,@#UBREAK  ;LOAD MICRO BREAK REG WITH SW0-7

                                ;CHECK THAT CP CAN TIME OUT TRAP
001056  012737  001074  000004          MOV     #TORET,@#ERRVEC ;LOAD TIMEOUT TRAP VECTOR
001064  005037  173000                  CLR     @#173000        ;ADDRESS 173000 ALWAYS TIMES OUT ON
                                                                ;DATIP/DATO BUS CYCLE
001070  000000                          HLT                     ;ERROR! FAILED TO TIME OUT TRAP
001072  000755                          BR      BEGIN           ;LOOP TEST
001074  022626                  TORET:  CMP     (6)+,(6)+       ;RESTORE THE STACK

                                ;TEST THAT THE STACK LIMIT REGISTER CAN BE REFERENCED
                                ;USING DATI, DATIP/DATO
001076  010701                  T0:     SCOPE
001100  012767  001122  176676          MOV     #T0A,ERRVEC     ;LOAD ERROR VECTOR
001106  005067  176674                  CLR     ERRVEC+2
001112  017737  177366  177774          MOV     @SLR,@#177774   ;REFERENCE STACK LIMIT REGISTER
001120  000403                          BR      T0B             ;GO TO NEXT TEST
001122  022626                  T0A:    POP2
001124  000000                          HLT                     ;ERROR! CANNOT REFERENCE STACK LIMIT REG,
001126  000763                          BR      T0              ;LOOP TEST IF ERROR

                                ;USING DATI, DATIP/DATOB

001130  012767  001146  176646  T0B:    MOV     #T0C,ERRVEC     ;LOAD ERROR VECTOR
001136  117777  177342  177342          MOVB    @SLR,@SLH       ;REFERENCE ODD BYTE
001144  000403                          BR      T0D
001146  022626                  T0C:    POP2
001150  000000                          HLT                     ;ERROR! CANNOT REFERENCE STACK LIMIT
001152  000766                          BR      T0B             ;USING BYTE INSTRUCTION.

001154  012767  000006  176622  T0D:    MOV     #6,ERRVEC       ;RESTORE ERROR TRAP VECTOR
001162  000400                          BR      T1              ;GO TO NEXT TEST

                                ;TEST THAT EACH BIT OF THE STACK LIMIT REGISTER BITS CAN BE SET
                                ;AND CLEARED, THIS TEST ROTATES A BIT THROUGH THE STACK LIMET REGISTER,

001164  010701                  T1:     SCOPE
001166  012702  000400                  MOV     #400,R2         ;LOAD TEST VALUE
001172  010277  177306          T1A:    MOV     R2,@SLR         ;LOAD TEST VALUE INTO STACK LIM, REG,
001176  017700  177302                  MOV     @SLR,R0         ;GET RESULT
001202  020002                          CMP     R0,R2           ;CHECK RESULT
001204  001401                          BEQ     .+4             ;BRANCH IF RESULT IS CORRECT
001206  000000                          HLT                     ;ERROR! INCORRECT RESULT, R2 HAS CORRECT
                                                                ;RESULT AND R0 HAS INCORRECT RESULT
```

```
001210  005077  177270                CLR     @SLR            ;CLEAR STACK LIM. REG.
001214  017700  177264                MOV     @SLR,R0         ;GET AND CHECK RESULT
001220  001401                        BEQ     .+4
001222  000000                        HLT                     ;ERROR! INCORRECT RESULT
001224  006302                        ASL     R2              ;SHIFT TEST VALUE
001226  103361                        BCC     T1A             ;BRANCH IF NOT DONE

                              ;THIS TEST INCREMENTS THE STACK LIMIT REGISTER

001230  010701                T2:     SCOPE
001232  005067  177252                CLR     TEMP
001236  005077  177242                CLR     @SLR            ;CLEAR STACK LIMIT REGISTER
001242  017700  177236        T2A:    MOV     @SLR,R0         ;GET RESULT
001246  020067  177236                CMP     R0,TEMP         ;CHECK RESULT
001252  001401                        BEQ     .+4
001254  000000                        HLT                     ;ERROR! STACK LIM. REG. HAS INCORRECT DATA
                                                              ;TEMP = CORRECT RESULT
001256  105277  177224                INCB    @SLH            ;INCREMENT VALUE IN STACK LIM. REG.
001262  105267  177223                INCB    TEMP+1          ;INCREMENT TEST VALUE
001266  001365                        BNE     T2A             ;BRANCH IF ALL VALUES NOT TESTED
```

```
                                ;TEST THAT RESET CLEARS THE STACK LIMIT REGISTER

001270  010701                  T3:   SCOPE
001272  012777  177777  177204        MOV     #-1,@SLR        ;PRESET SLR
001300  017700  177200              MOV     @SLR,R0         ;SAVE SLR
001304  022700  177400              CMP     #177400,R0      ;CHECK THAT SLR WAS LOADED
001310  001401                      BEQ     .+4
001312  000000                      HLT                     ;ERROR: SLR FAILED TO LOAD
001314  000005                      RESET                   ;RESET CLEARS SLR
001316  017700  177162              MOV     @SLR,R0         ;GET RESULT OF RESET AND CHECK RESULT
001322  001402                      BEQ     T4              ;GO TO NEXT TEST IF RESET CLEARED SLR
001324  000000                      HLT                     ;ERROR: RESET FAILED TO CLEAR SLR
001326  000760                      BR      T3              ;LOOP TEST IF ERROR

                                ;TEST THAT THE CLEAR INSTRUCTION CLEARS THE STACK LIMIT REGISTER

001330  010701                  T4:   SCOPE
001332  112777  177777  177146        MOVB    #-1,@SLH        ;PRESET ODD BYTE
001340  017700  177140              MOV     @SLR,R0         ;GET RESULT
001344  022700  177400              CMP     #177400,R0      ;CHECK RESULT
001350  001401                      BEQ     .+4
001352  000000                      HLT                     ;ERROR: SLR DID NOT PRESET
001354  005077  177124              CLR     @SLR
001360  017700  177120              MOV     @SLR,R0         ;GET RESULT OF CLEAR & BRANCH IF CLEAR
001364  001411                      BEQ     T5              ;GO TO NEXT TEST
001366  000000                      HLT                     ;ERROR: CLR INST FAILED TO CLEAR SLR
001370  000757                      BR      T4              ;LOOP TEST IF ERROR

        001410                      .=1410
                                ;TEST THAT AN OVERFLOW ERROR OCCURS FOR ALL STACK LIMIT REGISTER VALUES,
                                ;(PROVIDED CORE IS AVAILABLE).

001410  010701                  T5:   SCOPE
001412  012702  000010              MOV     #10,R2          ;INITIALIZE STACK VALUE AND
001416  012703  177400              MOV     #-400,R3        ;STACK LIMIT REGISTER VALUE

001422  062702  000400          T5A:  ADD     #400,R2         ;LOAD NEW STACK VALUE
001426  062703  000400              ADD     #400,R3         ;AND NEW STACK LIM. REG. VALUE
001432  005037  000000              CLR     @#0             ;CLEAR ADDRESS 0
001436  016706  177040              MOV     SPBOT,SP        ;INITIALIZE THE STACK POINTER

001442  012767  001704  176334  T5B:  MOV     #LIMX,ERRVEC    ;LOAD TIME OUT TRAP
001450  016204  177776              MOV     -2(2),R4        ;SAVE STACK LOCATIONS
001454  016205  177774              MOV     -4(2),R5        ;EXIT TEST IF EITHER INST TIMES OUT
001460  012767  001504  176316        MOV     #LIMA,ERRVEC    ;LOAD OVERFLOW VECTOR
001466  010206                      MOV     R2,SP           ;LOAD STACK POINTER
001470  010377  177010              MOV     R3,@SLR         ;LOAD STACK LIM. REG.
001474  016666  177770  177770        MOV     -10(6),-10(6)   ;REFERENCE LIMIT ADDRESS
001502  000401                      BR      .+4             ;SHOULD NOT HAVE TRAPPED
001504  000000                LIMA: HLT                     ;ERROR: REFERENCE TO LIMIT ADDRESS
                                                            ;CAUSED AN OVERFLOW
001506  010206                      MOV     R2,SP           ;REPOINT STACK POINTER
001510  012767  001526  176266        MOV     #LIMB,ERRVEC    ;REPOINT OVERFLOW VECTOR
```

```
001516  016666  177766  177766          MOV     -12(6),-12(6)   ;REFERENCE FIRST 'YELLOW' ADDRESS
001524  000000                          HLT                     ;ERROR! SHOULD HAVE TRAPPED
001526  005737  000000          LIMB:   TST     @#0             ;WAS IT A 'RED' OVERFLOW?
001532  001401                          BEQ     .+4
001534  000000                          HLT                     ;ERROR! A 'RED' OVERFLOW OCCURRED WHEN
                                                                ;A 'YELLOW' ADDRESS WAS REFERENCED.

001536  010206                          MOV     R2,SP           ;REPOINT STACK POINTER
001540  005037  000000                  CLR     @#0             ;CLEAR ADDRESS 0
001544  012767  001562  176232          MOV     #LIMC,ERRVEC    ;REPOINT OVERFLOW VECTOR
001552  016666  177730  177730          MOV     -50(6),-50(6)   ;REFERENCE LAST 'YELLOW' ADDRESS
001560  000000                          HLT                     ;ERROR! SHOULD HAVE TRAPPED
001562  005737  000000          LIMC:   TST     @#0             ;WAS IT A 'RED' OVERFLOW
001566  001401                          BEQ     .+4
001570  000000                          HLT                     ;ERROR! A 'RED' OVERFLOW OCCURRED WHEN
                                                                ;THE LAST 'YELLOW' ADDRESS WAS REFERENCED

001572  005037  000000                  CLR     @#0             ;CLEAR ADDRESS 0
001576  010206                          MOV     R2,SP           ;REPOINT THE STACK POINTER
001600  012767  001622  176176          MOV     #LIMD,ERRVEC    ;REPOINT THE OVERFLOW TRAP
001606  016267  177726  176674          MOV     -52(2),TEMP     ;GET 'RED' LOCATION
001614  005166  177726                  COM     -52(6)          ;REFERENCE 'RED' ADDRESS
001620  000000          LIMCC:  HLT                             ;ERROR! NO OVERFLOW TRAP WHEN 'RED' ADDRESS
                                                                ;WAS REFERENCED

001622  026267  177726  176660  LIMD:   CMP     -52(2),TEMP     ;WAS INSTRUCTION ABORTED?
001630  001401                          BEQ     .+4
001632  000000                          HLT                     ;ERROR! COM  -52(6) WAS ALLOWED TO CHANGE
                                                                ;A 'RED' ADDRESS.

001634  016762  176650  177726          MOV     TEMP,-52(2)     ;RESTORE 'RED' LOCATION IN ANY EVENT
001642  005706                          TST     %6              ;WAS STACK POINTER CHANGED TO 0?
001644  001401                          BEQ     .+4
001646  000000                          HLT                     ;'RED' OVERFLOW DID NOT ASSUME NEW STACK
001650  022737  001620  000000          CMP     #LIMCC,@#0      ;IS RETURN ADDRESS ON NEW STACK?
001656  001401                          BEQ     .+4
001660  000000                          HLT                     ;ERROR! RETURN ADDRESS NOT SAVED ON NEW
                                                                ;STACK

001662  005077  176616                  CLR     @SLR            ;GET READY TO GET NEW VALUES FOR TEST
001666  016706  176610                  MOV     SPBOT,SP        ;INITIALIZE THE STACK POINTER
001672  010462  177776                  MOV     R4,-2(2)        ;RESTORE STACK ADDRESS DATA
001676  010562  177774                  MOV     R5,-4(2)
001702  000647                          BR      T5A             ;GET NEW VALUES
001704  012767  000006  176072  LIMX:   MOV     #6,ERRVEC       ;RESTORE TIME OUT TRAP VECTOR


001712  005267  176562          END:    INC     ICNT
001716  026727  176556  002000          CMP     ICNT,#2000      ;HAVE 2000 PASSES BEEN COMPLETED
001724  001402                          BEQ     DONE
001726  000167  177074                  JMP     BEGIN
001732  012767  000007  175626  DONE:   MOV     #7,TPB          ;RING BELL
001740  105767  175620                  TSTB    TPS
001744  100375                          BPL     .-4
001746  013702  000042                  MOV     @#42,%2         ;GET DECTAPE MONITOR RETURN ADDRESS
001752  001404                          BEQ     DONE1           ;DO NOT RETURN IF (42)=0
001754  004712                          JSR     7,(2)           ;RETURN TO DECTAPE MONITOR
001756  000240                          NOP                     ;ACT11
```

DCKBFA  SYMBOL TABLE

```
BEGIN    001026        DISPLA = 177570        DONE     001732        DONE1    001764
END      001712        ERRVEC = 000004        HLT    = 000000        ICNT     000500
LIMA     001504        LIMB     001526        LIMC     001562        LIMCC    001620
LIMD     001622        LIMX     001704        PC     =%000007        POP2   = 022626
PRTY7  = 000340        PSW    = 177776        R0     =%000000        R1     =%000001
R2     =%000002        R3     =%000003        R4     =%000004        R5     =%000005
R6     =%000006        SCOPE  = 010701        SLH      000506        SLR      000504
SP     =%000006        SPBOT    000502        START    001010        SWR    = 177570
TEMP     000510        TORET    001074        TPB    = 177566        TPS    = 177564
T0       001076        T0A      001122        T0B      001130        T0C      001146
T0D      001154        T1       001164        T1A      001172        T2       001230
T2A      001242        T3       001270        T4       001330        T5       001410
T5A      001422        T5B      001442        UBREAK = 177770        .      = 001770
```

ERRORS DETECTED:  0

```
001760  000240                          NOP                     ;OVERLAY
001762  000240                          NOP                     ;AREA
001764  000167  177020          DONE11  JMP     START           ;RESTART

        000001                          .END
```

IDENTIFICATION

| | |
|---|---|
| PRODUCT CODE: | MAINDEC-11-DBKTG-A -D |
| PRODUCT NAME: | KT11-D EXERCISER |
| DATE CREATED: | NOV. 1, 1972 |
| MAINTAINER: | DIAGNOSTIC PROGRAMMING |
| AUTHOR: | ROBERT WHITTON |

ADVANCE COPY
This document subject to change without notice.

COPYRIGHT © 1972
DIGITAL EQUIPMENT CORPORATION

1,0 ABSTRACT

THIS PROGRAM IS AN INTERACTIVE EXERCISER FOR A PDP-11/40 EQUIPPED WITH THE KT11-D OPTION. IT PERFORMS A TEST OF INSTRUCTIONS AND CONCURRENT OPERATIONS OF I/O EQUIPMENT WHILE RELOCATING THRU MEMORY. IT PROVIDES NUMEROUS MODES OF TESTING, FROM 4K EXECUTION WITH THE KT11-D TURNED OFF AND ONLY KERNEL MODE IN USE, TO 128K EXECUTION WITH EACH USER PAGE MAPPED SEQUENTIALLY TO EVERY 4K BANK OF MEMORY. THIS PROGRAM IS NOT TO BE CONSIDERED A TOTAL CHECK OF THE SYSTEM. IF AN ERROR IS DETECTED IN AN I/O DEVICE, IT WILL PROBABLY BE NECESSARY TO CORRECT THE MALFUNCTION WITH THE RESPECTIVE DIAGNOSTIC FOR THAT DEVICE.

2,0 REQUIREMENTS

2,1 EQUIPMENT

PDP-11/40 STANDARD COMPUTER
KT11-D MEMORY MANAGEMENT OPTION

2,1,1 OPTIONAL HARDWARE THAT THE PROGRAM WILL EXERCISE

| | |
|---|---|
| MEMORY | UP TO 124 KW OF MEMORY-DOES NOT HAVE TO BE CONTIGUOUS, BUT BLOCKS OF LESS THAN 4KW WILL NOT BE USED |
| RF11 | DISK |
| RK11 | DISK |
| TC11 | DECTAPE-TRANSPORT ONE(1) |
| KW11-L | LINE CLOCK |
| KL11 | ASR33 OR ASR35 TELEPRINTER |
| LP11 | LINE PRINTER |

2,2 STORAGE

THIS PROGRAM USES MEMORY FROM 00000 TO 17760.

3,0 LOADING PROCEDURE

PROCEDURE FOR NORMAL ABSOLUTE TAPES SHOULD BE FOLLOWED.

4,0 STARTING PROCEDURE AND SWITCH SETTINGS

4,1 NORMAL STARTING PROCEDURE

LOAD STARTING ADDRESS 200.
SET DESIRED MEMORY MANAGEMENT SELECTION SWITCHES (SEE 4,2)-ALL DOWN FOR WORST CASE TESTING.
PRESS START.

4,1 NORMAL STARTING PROCEDURE (CONTINUED)

THE PROGRAM WILL IMMEDIATELY HALT. AT THE HALT, SET THE DESIRED DEVICE SELECTION SWITCHES (SEE 4,3) AND THE DESIRED DYNAMIC SWITCHES (SEE 5,1,2).
PRESS CONTINUE.
THE PROGRAM WILL RING THE BELL (UNLESS THE TTY OUTPUT IS SELECTED) AT THE END OF EACH BANK. IF SWITCHES 0,1 AND 2 WERE ALL DOWN WHEN START WAS PRESSED (SELECTING THE USE OF 4K PHYSICAL ADDRESS SPACE AS 32K VIRTUAL ADDRESS SPACE-SEE 5,3,1) AN ASTERISK WILL BE TYPED AT THE END OF A FULL PASS THRU ALL MEMORY (UNLESS THE TTY OUTPUT IS SELECTED).

4,2 MEMORY MANAGEMENT SELECTION SWITCHES (INITIAL SWITCH REGISTER SETTINGS).

THE SWITCHES SET AT STARTUP DETERMINE THE WAY IN WHICH MEMORY IS MAPPED AND EXERCISED:

SW0=1 OR UP---INHIBIT THE KT11-D (SR0<0> WILL NOT BE SET AT ALL)
SW1=1 OR UP---INHIBIT USE OF USER MODE.
(ALSO INHIBITS 4K AS 32K)
SW2=1 OR UP---INHIBIT 4K AS 32 K (ALSO INHIBITED IF EITHER SW0 OR SW1 IS SET)-SEE SECTION 5,3,1 FOR EXPLANATION
SW3=1 OR UP---INHIBIT VARIABLE CORE EXPANSION
=0 OR DOWN-CORE EXPAND UNLESS SW0, 1 AND 2 ARE ALL DOWN
(IN WHICH CASE 4K AS 32K IS RUN INSTEAD)

4,3 DEVICE SELECTION SWITCHES

THE DEVICE SELECTION SWITCHES ARE SET AT THE FIRST (AND ONLY) HALT. EACH SWITCH, IF SET, INHIBITS A SINGLE I/O DEVICE FROM BEING EXERCISED. IF A DEVICE DOES NOT EXIST, THE CORRESPONDING INHIBIT SWITCH MUST BE SET. (THIS WILL BE CHANGED IN A FUTURE VERSION).

SW0=1 OR UP---INHIBIT TTY OUTPUT
SW3=1 OR UP---INHIBIT RK11 DISK
SW4=1 OR UP---INHIBIT LINE CLOCK
SW5=1 OR UP---INHIBIT RF11 DISK
SW6=1 OR UP---INHIBIT TC11 DECTAPE
SW7=1 OR UP---INHIBIT LINE PRINTER (USE SA318 IF LP11 IS SELECTED)

4,4 RESTART PROCEDURE

USING RESTART ADDRESS 300 THE SWITCH REGISTER SETTINGS GIVEN PREVIOUSLY ARE USED (FOR BOTH MEMORY MANAGEMENT SELECTION AND DEVICE SELECTION), NO HALT OCCURS AFTER START IS PRESSED.

5, OPERATING PROCEDURE

5,1 OPERATIONAL SWITCH SETTINGS

5,1,1 BASIC SWITCH SETTINGS-STARTUP

SEE SECTIONS 4,2 AND 4,3 FOR THE BASIC SWITCH SETTINGS USED AT STARTUP. THOSE SWITCHES ARE NOT RECHECKED AFTER THEY ARE INITIALLY STORED.

5,1,2 DYNAMIC SWITCH SETTINGS

THE FOLLOWING SWITCHES ARE RECHECKED PERIODICALLY DURING PROGRAM EXECUTION:

SW15=1 OR UP---HALT ON ERROR
SW14=1 OR UP---SCOPE LOOP
SW13=1 OR UP---INHIBIT PRINT OUT
SW12=1 OR UP---INHIBIT TRACE TRAPPING
SW11=1 OR UP---INHIBIT SUB-PROGRAM ITERATION AND INHIBIT TESTS WHICH USE ALL COMBINATIONS OF NUMBERS
SW10=1 OR UP---INHIBIT PROCESSOR TEST (ONCE SET, PROCESSOR TEST IS PERMANENTLY INHIBITED)

5,2 SUBROUTINE ABSTRACTS

5,2,1 SCOPE

THIS SUBROUTINE CALL IS PLACED BETWEEN EACH SUBTEST. IT RECORDS THE STARTING ADDRESS OF EACH SUBTEST AS IT IS BEING ENTERED. IF A SCOPE LOOP IS REQUESTED, IT WILL JUMP TO THE START OF THE SUBTEST THAT THE SCOPE LOOP IS REQUESTED FOR. IF A SCOPE LOOP IS NOT REQUESTED, THERE WILL BE 256 ITERATIONS ON THAT SUBTEST BEFORE THE NEXT SUBTEST IS ENTERED. SWITCH 11 ON A 1 INHIBITS ITERATION OF SUBTESTS.

5,2,2 HLT

THIS EMT CALLS THE SUBROUTINE PRINT, WHICH PRINTS OUT THE LOCATION COUNTER AT THE TIME OF FAILURE, THE CONTENTS OF THE PROCESSOR STATUS REGISTER, AND THE CONTENTS OF THE CURRENT BANK COUNTER. NOTE THAT THE LOCATION COUNTER WILL BE THE VIRTUAL ADDRESS OF THE HLT PLUS TWO.

5,2,3 TRAPCATCHER

THIS IS A SERIES OF INSTRUCTIONS STARTING AT LOCATION 0 DESIGNED TO DETECT AND ISOLATE UNEXPECTED TRAPS AND INTERRUPTS TO THE TRAP AND INTERRUPT VECTOR AREA OF MEMORY.

EACH VECTOR ENTRANCE ADDRESS IS LOADED WITH THE ADDRESS OF THE NEXT LOCATION, THE NEXT LOCATION IS LOADED WITH A HALT (000000), THUS AN ILLEGAL TRAP OR INTERRUPT WILL CAUSE A HALT AT THE TRAP LOCATION PLUS TWO.

IF A HALT OCCURS IN THE TRAP OR INTERRUPT AREA EXAMINE KERNEL REGISTER SIX. IT WILL CONTAIN THE CURRENT STACK ADDRESS, THE CONTENTS OF THE CURRENT STACK ADDRESS IS THE VIRTUAL PC AT THE TIME THE TRAP OR INTERRUPT OCCURRED.

5,2,4 EMTSRV (EMT HANDLER)

THIS ROUTINE DECODES THE EMT CALLS AND PASSES CONTROL TO THE CORRECT SERVICE ROUTINE. THE ROUTINES HANDLED BY EMT CALLS ARE PRINT (HLT CALL) AND EOBSRV (EOB CALL).

5,2,6 EOBSRV (END OF BANK SERVICE)

THE VARIOUS EXECUTION OPTIONS FOR THIS EXERCISER REQUIRE SPECIAL HANDLING WHEN THE END OF THE PROCESSOR TESTS IS REACHED IN A BANK. THIS SERVICE ROUTINE PERFORMS THE VARIOUS MAPPING FUNCTIONS, DEPENDING UPON THE INITIAL SWITCH REGISTER SETTINGS.

5,2,7 BEGINX (CORE EXPANSION SPECIAL HANDLER)

WHEN CORE EXPANSION IS UTILIZED, A NUMBER OF SPECIAL ACTIONS MUST BE TAKEN AT THE BEGINNING OF EACH BANK, THE $SCOPE ROUTINE VECTOR IS LOADED TO POINT TO THE NEW BANK, AND IF TC11 AND RF11 CODE AND BUFFER RELOCATION IS ALLOWED.

5,2,9 PFAIL (POWER FAIL)

IN THIS VERSION THE POWER FAIL ROUTINE IS NOT OPERABLE.

5,2,11 TYOUT (TTY OUTPUT)

THIS ROUTINE OUTPUTS A COUNT PATTERN IN THE INTERRUPT MODE TO THE TELEPRINTER.

5,2,12 RFSTART (RF11 DISK)

THIS ROUTINE PERFORMS A WRITE AND A WRITE CHECK OF THE DISK. THE DATA THAT IS WRITTEN ON THE DISK IS A PART OF THE TEST PROGRAM CODE THAT IS NEVER MODIFIED. THIS SEGMENT OF CORE IS WRITTEN IN CONTIGUOUS BLOCKS THRU THE DISK MEMORY. AFTER THE TOTAL DISK(S) HAS BEEN WRITTEN, A WRITE CHECK IS USED TO VERIFY THAT THE DATA HAS BEEN WRITTEN CORRECTLY ON THE DISK. NOTE THAT NO "DATI" ARE USED IN EXERCISING THE DISK (DATA IS NOT TRANSFERRED INTO MEMORY). THERE IS A LOCATION IN THE PROGRAM THAT IF MODIFIED WILL ALLOW EXERCISING UP TO EIGHT DISKS.

5,2,13 ENDZ (TC11 END ZONE HANDLER)

THIS ROUTINE IS PART OF THE TC11 SERVICE CODE, IT DRIVES THE DECTAPE INTO THE FORWARD OR REVERSE END ZONE, THEN REVERSES IT, IT ALSO DOES THE NECESSARY SETUP TO BEGIN READING OR WRITING THE TAPE.

5,2,14 REGEN (TC11 WRITE BUFFER REGENERATE ROUTINE)

THE TC11 CODE WRITES THE ENTIRE DECTAPE GOING FORWARD, THEN READS IT IN REVERSE. THE BUFFER IS REGENERATED BEFORE WRITING THE TAPE, AND IS CLEARED OUT ONCE THE ENTIRE TAPE HAS BEEN WRITTEN. THIS ROUTINE REGENERATES THE WRITE BUFFER.

5,2,15 RBN (TC11 READ BLOCK NUMBER SERVICE ROUTINE)

AT THE END OF EACH "BLOCK NUMBER FOUND" INTERRUPT, THIS ROUTINE IS ENTERED (UNLESS END ZONE IS BEING SEARCHED FOR), IT CHECKS FOR THE CORRECT SEQUENCE OF BLOCK NUMBERS, THEN SETS UP THE TC11 TO WRITE A BLOCK IF THE TAPE IS TRAVELLING FORWARD. IF IT IS GOING IN REVERSE, THE ROUTINE CHECKS TO SEE IF DATA IS STILL BEING CHECKED FROM A PREVIOUS READ. IF IT'S NOT, THE ROUTINE SETS UP TO READ A BLOCK. IF DATA IS STILL BEING CHECKED FROM BEFORE, IT SIMPLY DOES ANOTHER READ BLOCK NUMBER.

5,2,16 NXTBLK (TC11 READ BLOCK AND WRITE BLOCK SERVICE ROUTINE)

WHEN A READ BLOCK OR A WRITE BLOCK OPERATION IS COMPLETED, THIS ROUTINE IS ENTERED. IT CHECKS THE ERROR BIT, THEN SETS UP A CALL TO CHECK DATA IF DATA WAS JUST READ IN. THE ROUTINE ALSO SETS UP A READ BLOCK NUMBER OPERATION.

5,2,17 TCCK (TC11 CHECK DATA ROUTINE)

WHEN A READ BLOCK OPERATION HAS BEEN COMPLETED, THIS ROUTINE IS CALLED VIA A PRIORITY INTERRUPT REQUEST AT LEVEL 5. THE ENTIRE BUFFER IS CHECKED, AND THE CONTENTS OF THE BUFFER IS ALTERED AS THE CHECK PROGRESSES. THUS, IF A READ BLOCK OPERATION DOES NOT ACTUALLY READ IN ANY DATA, THE DATA CHECK ROUTINE WILL FIND BAD DATA INSTEAD OF SEEING GOOD DATA FROM AN EARLIER READ.

5,2,18 LCLK (LINE CLOCK)

THIS TEST OF THE LINE CLOCK IS IN THE INTERRUPT MODE, IF OPERATING CORRECTLY THE SYSTEM I/O WILL RUN AT FULL SPEED FOR 55 SECONDS, AND THEN ALL I/O AT LEVEL FOUR OR LESS (AND THE PROCESSOR TESTS) WILL STALL FOR 5 SECONDS, TIMES GIVEN ARE BASED ON 60 CYCLES AS THE LINE FREQUENCY.

5,2,19 LP1 (LINE PRINTER)

THIS ROUTINE OUTPUTS TO THE LINE PRINTER IN THE FLAG MODE WHILE FILLING THE BUFFER, AND IN THE INTERRUPT MODE WHILE THE BUFFER IS BEING PRINTED.

5,2,20 RKSTART (RK-11 DISK)

THIS ROUTINE PERFORMS A WRITE AND WRITE CHECK OF THE DISK, THE DATA THAT IS WRITTEN ON THE DISK IS PART OF THE TEST PROGRAM CODE THAT IS NEVER MODIFIED, THIS SEGMENT OF CORE IS WRITTEN IN CONTIGUOUS BLOCKS THRU THE DISK MEMORY, AFTER THE TOTAL DISK HAS BEEN WRITTEN, A WRITE CHECK IS USED TO VERIFY THAT DATA HAS BEEN WRITTEN CORRECTLY ON THE DISK.

5,2,22 CORE EXPANSION (DET1)

THIS ROUTINE IS CONTROLLED BY SWITCH 5, IF CALLED, THE PROCESSOR MAINLINE CODE WILL EXPAND TO THE MAXIMUM MEMORY THAT IS AVAILABLE (UP TO 28K), THE ROUTINE DETERMINES THE MAXIMUM MEMORY SIZE BY DOING A "DATO" TO A LOCATION IN EACH BANK, IF THE BANK DOES NOT EXIST, A TIMEOUT WILL OCCUR, AN IMAGE OF BANK 0 IS THEN TRANSFERRED TO EACH EXISTING BANK, THE CODE IN EACH BANK EXCEPT THE LAST IS MODIFIED TO CHANGE THE END OF BANK CALL TO A JUMP TO BEGINX (CORE EXPANSION SPECIAL HANDLER) IN THE NEXT BANK.

THE LISTING SHOWS ONLY THE CODE FOR BANK ZERO, WHEN AN ERROR OCCURS THAT IS NOT IN BANK ZERO, IGNORE THE BANK BITS OF THE PRINT OUT AND USE THE LISTING FOR BANK ZERO.

5.3 PROGRAM AND/OR OPERATOR ACTION

5.3.1 PROCESSOR TEST EXECUTION - 4K AS 32K

IF SWITCHES 0, 1, AND 2 ARE ALL DOWN (=0) AT STARTUP, THE PROCESSOR TEST WILL BE EXECUTED TREATING EACH 4K BANK AS 32K OF VIRTUAL ADDRESS SPACE, THE FOLLOWING DETAILS THIS MODE OF OPERATION,

USER PAGE 0 IS FIRST MAPPED RW, BANK 0, AND ALL OTHER USER PAGES ARE MAPPED NON-RESIDENT, THE PROCESSOR TESTS ARE EXECUTED IN USER THRU USER PAGE 0, WHEN DONE, USER PAGE 0 IS CHANGED TO NON-RESIDENT, AND USER PAGE 1 IS MAPPED RW, BANK 0, THE PC IS CHANGED TO ADDRESS THE START OF THE PROCESSOR TESTS THRU PAGE 1, AND ANOTHER PASS THRU THE PROCESSOR TESTS IS EXECUTED, AT THE END OF THIS PASS, USER PAGE 2 IS MAPPED RW, BANK 0, AND USER PAGE 1 IS MADE NON-RESIDENT, THE PC IS AGAIN CHANGED, THIS TIME TO ACCESS USER PAGE 2, AND THE PROCESSOR TESTS ARE EXECUTED THRU USER PAGE 2, THIS CYCLE IS REPEATED FOR THE REMAINING USER PAGES, MAPPING EACH IN TURN TO BANK 0 AND CHANGING THE PC TO EXECUTE THRU THE ONE CURRENTLY MAPPED, WHEN THE PASS USING USER PAGE 7 IS COMPLETED, A SEARCH IS MADE FOR THE NEXT 4K BANK OF MEMORY, WHEN A BANK IS FOUND, THE PROGRAM IS COPIED INTO THAT BANK FROM BANK 0, USER PAGE 0 IS MAPPED TO THE NEW BANK, AND THE PC IS CHANGED TO EXECUTE THRU USER PAGE 0, THE PREVIOUS CYCLE IS REPEATED, BUT THIS TIME EACH USER PAGE IS MAPPED IN TURN TO THE NEW BANK, ONCE EXECUTION THRU USER PAGE 7 IS COMPLETED, A SEARCH IS MADE FOR THE NEXT BANK, THE PREVIOUS BANK IS CLEARED (EXCEPT FOR THE LOADER), AND THE PROGRAM IS COPIED FROM BANK 0 INTO THE CURRENT BANK, THE CYCLE REPEATS UNTIL THE EXTERNAL BANK IS REACHED, AT WHICH POINT USER 0 IS MAPPED BACK TO BANK 0 AND THE PROCESS STARTS AGAIN,

5.3.2 PROCESSOR TEST EXECUTION - CORE EXPANSION

IF SWITCH 0, 1, OR 2 IS UP AND SW5 IS DOWN AT STARTUP, THE PROCESSOR TESTS WILL BE CORE EXPANDED THRU ALL AVAILABLE MEMORY UP TP 28K. THR ROUTINE DET1 DOES THIS CORE EXPANSION, COPYING BANK 0 INTO EACH OF THE OTHER BANKS. THE EMT CALL AT THE END OF EACH BANK (EOB) WHICH CALLS THE END OF BANK SERVICE ROUTINE IS CHANGED TO A JUMP TO BEGINX IN THE NEXT BANK. THE EOB CALL IN THE LAST BANK IS LEFT ALONE. IF SWITCHES 0 AND 1 WERE BOTH DOWN AT STARTUP, USER PAGES 0 THRU 6 ARE MAPPED SO THAT THE PHYSICAL AND VIRTUAL ADDRESSES CORRESPOND, AND THE PROCESSOR TESTS ARE THEN RUN IN USER. IF SW0 WAS DOWN BUT SW1 WAS SET, KERNEL PAGES 0-6 ARE MAPPED SO THAT THE PHYSICAL AND VIRTUAL ADDRESSES ARE THE SAME, AND THE PROCESSOR TESTS ARE THEN RUN IN KERNEL MODE. IF SW0 WAS SET, ORDINARY CORE EXPANSION IS RUN WITH NO SPECIAL MAPPING REQUIRED (KT11-D IS TURNED OFF).

5.3.3 PROCESSOR TEST EXECUTION - BANK 0 ONLY

IF SW0, 1 OR 2 IS UP AND SW5 IS UP AT STARTUP, ONLY BANK 0 IS UTILIZED. IN THIS CASE, IF SW0 AND SW1 WERE DOWN THE PROCESSOR TESTS ARE EXECUTED IN USER, WITH USER PAGE 0 MAPPED TO BANK 0. IF SW0 WAS DOWN AND SW1 WAS UP, THE PROCESSOR TESTS ARE EXECUTED IN KERNEL, WITH KERNEL PAGE 0 MAPPED TO BANK 0. IF SW0 WAS UP, THE KT11-D IS TURNED OFF AND THE PROCESSOR TESTS ARE EXECUTED IN KERNEL MODE OR USER MODE (DEPENDING ON SW1) IN BANK 0 ONLY.

6,0 ERRORS

6,1 ERROR PRINTOUT

PRINTOUTS ARE IN AN EXTENDED VERSION OF THE STANDARD FORMAT, USING THREE WORDS. THE FIRST WORD IS THE OCTAL VALUE OF THE VIRTUAL PC+2 OF THE DETECTED ERROR. THE SECOND WORD IS THE CONTENTS OF THE PROCESSOR STATUS REGISTER WHEN THE ERROR WAS DETECTED. THE THIRD IS THE TOP 12 BITS OF THE 18-BIT ADDRESS OF THE BANK BEING CURRENTLY USED FOR EXECUTION OF THE PROCESSOR TEST. THE FOURTH IS RETURN WHICH IS THE RETURN ADDRESS IN THE CURRENT BANK OF MEMORY.
TO GET THE STARTING ADDRESS OF THE CURRENT BANK SIMPLY APPEND TWO ZEROS TO THE END OF THE OCTAL VALUE PRINTED OUT (I.E. 007400 INDICATES THE BANK BEGINNING AT PHYSICAL ADDRESS 740000).

6,2 ERROR RECOVERY

IN GENERAL, TEST FAILURES WILL PRINTOUT AN ERROR MESSAGE AND CONTINUE. IF THE "HALT ON ERROR" SWITCH IS SET, HITTING CONTINUE WILL RECOVER. IF THE PROGRAM HANGS UP IN A LOOP, THE ERROR IS LIKELY TO BE A SIGNAL WHICH WAS NEVER RECEIVED. IF A HALT OCCURS IN THE TRAP AND VECTOR AREA THE PROGRAM MUST BE RESTARTED. IF THE PROGRAM HALTS IN THE MAIN FLOW, CONSULT THE LISTING IF NO MESSAGE IS TYPED OUT. FOR TTY READER AND HSR, TAPE MUST BE REPOSITIONED TO LEADER BEFORE RESTARTING THE TEST.

6,3 FINDING WHICH PROCESSOR TEST WAS BEING EXECUTED WHEN AN ERROR OCCURRED

SOME ERRORS ARE DEPENDENT ON THE PROCESSOR TEST BEING RUN (SUCH AS LATENCY ERRORS WHICH ONLY SHOW UP IN WORST-CASE PROCESSOR TIMING), THE SCOPE ROUTINE CONTAINS A LOCATION CALLED "RETURN" WHICH STORES THE STARTING ADDRESS OF THE PROCESSOR TEST CURRENTLY BEING EXECUTED. NOTE THAT THE SCOPE ROUTINE IS EXECUTED IN USER MODE IF SW1 IS DOWN AT STARTUP, AND IS THEREFORE RELOCATED WITH THE PROCESSOR TESTS. THUS, TO DETERMINE WHICH PROCESSOR TEST WAS BEING EXECUTED WHEN A FAILURE OCCURRED, FIRST CHECK THE CONTENTS OF CURBNK IN BANK 0. THIS LOCATION CONTAINS THE ADDRESS OF THE CURRENT PHYSICAL BANK, SHIFTED RIGHT 6 PLACES, BY APPENDING 2 ZEROES TO IT, YOU HAVE THE 18-BIT ADDRESS OF THE CURRENT BANK OF MEMORY. ADD TO THIS THE ADDRESS OF RETURN IN BANK 0 AND YOU HAVE THE ADDRESS OF RETURN IN THE CURRENT BANK OF MEMORY, THE CONTENTS OF RETURN IN THE CURRENT BANK OF MEMORY IS THE VIRTUAL ADDRESS OF THE START OF THE CURRENT PROCESSOR TEST.

7,0 RESTRICTIONS

PROGRAM MUST BE LOADED INTO LOWER 4K OF MEMORY.

THE INHIBIT SWITCHES MUST ONLY BE SET FOR ALL DEVICES THAT ARE PART OF THE SYSTEM BUT WHICH YOU DO NOT WISH TO RUN.

IF THE LINE PRINTER IS USED, STARTING ADDRESS 310 MUST BE USED.

8.0 MISCELLANEOUS

8.1 EXECUTION TIME

EXECUTION TIME VARIES WITH THE AMOUNT OF MEMORY, THE TYPES OF MEMORY, THE DEVICES RUN, AND THE OPTIONAL MODES OF EXECUTION USED.

A PASS RUN WITH CORE EXPANSION AND 4K AS 32K RELOCATION BOTH INHIBITED TAKES LESS THAN 10 SECONDS (RUNNING NO I/O).

A PASS RUN WITH 4K AS 32K, IN CORE MEMORY WITH NO I/O, TAKES ABOUT 15 SECONDS PER 4K BANK. (AN ASTERISK IS PRINTED AT THE END OF A FULL PASS, AND THE BELL IS RUNG AT THE END OF EACH 4K BANK).

A PASS RUN WITH 4K AS 32 WITH TAKES ABOUT 1 MINUTE PER 4K BANK.

8.2 STACK POINTERS

THE KERNEL STACK POINTER IS INITIALIZED TO 17760.

THE USER STACK POINTER IS INITIALIZED TO 400. IT IS RELOCATED THRU ALL USER PAGES AND TO EVERY 4K BANK IF THE 4K AS 32K MODE OF EXECUTION IS RUN.

8.3 MONITORING PHYSICAL AND VIRTUAL ADDRESSES

DURING EXECUTION OF 4K AS 32K, IT IS HELPFUL TO SET THE ADDRESS SELECTOR TO PROGRAM PHYSICAL AND THE DATA SELECTOR TO DATA PATHS. IF THIS IS DONE, THE ADDRESS LIGHTS WILL INDICATE THE CURRENT PHYSICAL ADDRESSES WHILE THE DATA LIGHTS WILL SHOW THE CURRENT VIRTUAL ADDRESSES (SINCE THEY ARE USED AS DATA A GREAT DEAL OF THE TIME).

9,0 PROGRAM DESCRIPTION

THIS MEMORY MANAGEMENT EXERCISER IS DESIGNED TO RUN BACKGROUND PROCESSOR TESTS AND FOREGROUND CONCURRENT I/O WITH MEMORY MANAGEMENT UTILIZED IN ANY OF SEVERAL DIFFERENT MODES, THE VARIOUS MODES AVAILABLE FOR UTILIZING MEMORY MANAGEMENT ARE INCLUDED TO AID IN FAULT ISOLATION BY PROVIDING A SERIES OF STEPS FROM SIMPLE TO COMPLEX, MEMORY MANAGEMENT CAN BE LEFT TURNED OFF AND THE PROCESSOR TESTS CAN STILL BE RUN IN 4K ONLY OR CORE EXPANDED UP TO 28K, WITH DECTAPE AND RF11 DISK OPERATING IN BANK 0 OR RELOCATED TO THE CURRENT BANK, WITH MEMORY MANAGEMENT ON, THE PROGRAM CAN BE RUN USING ONLY 4K, WITH EVERYTHING MAPPED IN KERNEL SPACE OR WITH USER AND KERNEL BOTH USED, AT THE NEXT LEVEL OF COMPLEXITY, CORE EXPANSION CAN BE RUN WITH MEMORY MANAGEMENT ON, USING KERNEL ONLY OR USING ALL BOTH MODES AS DESIRED, FINALLY, ALL AVAILABLE MEMORY (IN 4K PIECES) CAN BE UTILIZED BY RUNNING 4K AS 32K,
THERE IS NO MONITOR IN THE CONVENTIONAL SENSE, EACH DEVICE THAT IS TO BE EXERCISED HAS ITS OWN STAND ALONE ROUTINE THAT OPERATES IN THE INTERRUPT MODE, THESE ROUTINES NEED NO SUPERVISION OR MONITORING AFTER THEY ARE INITIATED, THERE IS A PRIMER AREA THAT CHECKS THE SWITCH REGISTER TO SEE WHAT DEVICES ARE TO BE INITIATED, IT SETS THE INTERRUPT ENABLE BIT IN THE DEVICE STATUS REGISTER, INITIALIZES THE DATA PATTERN, AND INITIATES AN OPERATION TO RAISE DATA FLAGS ON DEVICES THAT CAN NOT INITIATE THEM THEMSELVES, THE PRIMER CODE THEN ENTERS THE KT11-D SETUP CODE,
THE RF11 AND TC11 PRIMER CODE IS IN WITH THE KT11-D SETUP CODE SINCE THEY REQUIRE CERTAIN PARTS OF THE KT11-D CODE TO BE RUN FIRST, AFTER MEMORY MANAGEMENT IS TURNED ON, EXECUTION OF THE BACKGROUND PROCESSOR TESTS BEGINS, AND THE I/O DEVICES ARE SERVICED WHEN THEY INTERRUPT,

```
                        ;
                        ;COPYRIGHT 1972, DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS. 01754
                        ;PDP11/40 SYSTEM EXERCISER, WITH KT11-D ... TTY,PC11,KW11-L
                        ;LP11,RF11,TC11
                        ;TEST SIMULTANEOUS RUNNING OF I/O, WITH PROCESSOR INSTRUCTION TEST AND
                        ;WITH TRACE BIT ENABLED TO BE CONSIDERED MAINLINE CODE

                        ;I/O RUNS IN KERNEL MODE
                        ;CPU TESTS RUN IN USER MODE UNLESS INHIBITED BY SR SETTINGS
                        ;KT11-D IS UTILIZED

                        ;(R6) IS THE STACK POINTER
                        ;((R6)) IS THE PC+2 OF LOCATION WHERE THE TRAP ORIGINATED
                        ;FOR NORMAL OPERATION RUN WITH ALL SWITCHES DOWN
                        ;SA = 200
                        ;RESTART = 300 (SR SETTINGS PREVIOUSLY MADE ARE USED)

                        ;AT STARTUP, SR SETTINGS ARE:
                        ;SR 0=1 OR UP ... RUN WITHOUT KT11-D
                        ;SR 1=1 OR UP ... RUN ALL IN KERNEL MODE (INHIBITS RUNNING 4K AS 32K)
                        ;SR 2=1 OR UP ... INHIBIT RUNNING 28K USER KT11-D FROM EVERY 4K
                        ;                 ;BANK (ALLOW NORMAL CORE EXPANSION)
                        ;SR 5=1 OR UP...;INHIBIT VARIABLE CORE EXPANSION

                        ;AT HALT, SR SETTINGS ARE:
                        ;SR 15=1 OR UP..;HALT ON ERROR
                        ;SR 14=1 OR UP..;SCOPE LOOP
                        ;SR 13=1 OR UP..;INHIBIT PRINT OUT
                        ;SR 12=1 OR UP..;INHIBIT TRACE TRAPPING
                        ;SR 11=1 OR UP..;INHIBIT SUB-PROGRAM ITERATION AND INHIBIT TESTS WHICH
                        ;               ;USE ALL COMBINATIONS OF NUMBERS
                        ;SR 10=1 OR UP..;INHIBIT PROCESSOR TEST

                        ;SPECIAL DELETE SWITCHES-SET RESPECTIVE SWITCH TO A 1 TO INHIBIT
                        ;INITIATION OF DEVICE
                        ;SW 0=1 INHIBIT TTY OUTPUT
                        ;SW 3=1 INHIBIT RK11 DISK
                        ;SW 4=1 INHIBIT LINE CLOCK
                        ;SW 5=1 INHIBIT RF11 DISK
                        ;SW 6=1 INHIBIT TC11 DECTAPE
                        ;SW 7=1 INHIBIT LINE PRINTER

                        ;DEFINITIONS
000240                          NOP=240                 ;SYSTEM NULL OPERATION
104400                          SCOPE=TRAP              ;TRAP USED SCOPE LOOP AND ITERATION
000410                          TCSR=TTYCSR
000412                          TDBR=TTYDBR
177570                          SR=177570
177776                          PSR=177776
104006                          HLT=104006              ;ERROR PRINTOUT CALL
104010                          EOB=104010              ;END OF BANK CALL
000000                          R0=%0
000001                          R1=%1
```

```
                000002                  R2=%2
                000003                  R3=%3
                000004                  R4=%4
                000005                  R5=%5
                000006                  SP=%6
                000006                  R6=SP
                000007                  PC=%7

                                        .SBTTL MAIN
                                ;TRAP CATCHER
                000000                  .=0
000000  000002                  .+2                     ;TRAP ENTRANCE
000002  000000                  HALT                    ;TRAPPED TO PREVIOUS LOCATION
000004  000006                  .+2                     ;TRAP ENTRANCE
000006  000000                  HALT                    ;TRAPPED TO PREVIOUS LOCATION
000010  000012                  .+2                     ;TRAP ENTRANCE
000012  000000                  HALT                    ;TRAPPED TO PREVIOUS LOCATION
000014  000016                  .+2                     ;TRAP ENTRANCE
000016  000000                  HALT                    ;TRAPPED TO PREVIOUS LOCATION
000020  000022                  .+2                     ;TRAP ENTRANCE
000022  000000                  HALT                    ;TRAPPED TO PREVIOUS LOCATION
000024  000026                  .+2                     ;TRAP ENTRANCE
000026  000000                  HALT                    ;TRAPPED TO PREVIOUS LOCATION
000030  000032                  .+2                     ;TRAP ENTRANCE
000032  000000                  HALT                    ;TRAPPED TO PREVIOUS LOCATION
000034  000036                  .+2                     ;TRAP ENTRANCE
000036  000000                  HALT                    ;TRAPPED TO PREVIOUS LOCATION
000040  000042                  .+2                     ;TRAP ENTRANCE
000042  000000                  HALT                    ;TRAPPED TO PREVIOUS LOCATION
000044  000046                  .+2                     ;TRAP ENTRANCE
000046  000000                  HALT                    ;TRAPPED TO PREVIOUS LOCATION
000050  000052                  .+2                     ;TRAP ENTRANCE
000052  000000                  HALT                    ;TRAPPED TO PREVIOUS LOCATION
000054  000056                  .+2                     ;TRAP ENTRANCE
000056  000000                  HALT                    ;TRAPPED TO PREVIOUS LOCATION
000060  000062                  .+2                     ;TRAP ENTRANCE
000062  000000                  HALT                    ;TRAPPED TO PREVIOUS LOCATION
000064  000066                  .+2                     ;TRAP ENTRANCE
000066  000000                  HALT                    ;TRAPPED TO PREVIOUS LOCATION
000070  000072                  .+2                     ;TRAP ENTRANCE
000072  000000                  HALT                    ;TRAPPED TO PREVIOUS LOCATION
000074  000076                  .+2                     ;TRAP ENTRANCE
000076  000000                  HALT                    ;TRAPPED TO PREVIOUS LOCATION
000100  000102                  .+2                     ;TRAP ENTRANCE
000102  000000                  HALT                    ;TRAPPED TO PREVIOUS LOCATION
000104  000106                  .+2                     ;TRAP ENTRANCE
000106  000000                  HALT                    ;TRAPPED TO PREVIOUS LOCATION
000110  000112                  .+2                     ;TRAP ENTRANCE
000112  000000                  HALT                    ;TRAPPED TO PREVIOUS LOCATION
000114  000116                  .+2                     ;TRAP ENTRANCE
000116  000000                  HALT                    ;TRAPPED TO PREVIOUS LOCATION
000120  000122                  .+2                     ;TRAP ENTRANCE
000122  000000                  HALT                    ;TRAPPED TO PREVIOUS LOCATION
000124  000126                  .+2                     ;TRAP ENTRANCE
```

```
000126  000000          HALT            ;TRAPPED TO PREVIOUS LOCATION
000130  000132          ,+2             ;TRAP ENTRANCE
000132  000000          HALT            ;TRAPPED TO PREVIOUS LOCATION
000134  000136          ,+2             ;TRAP ENTRANCE
000136  000000          HALT            ;TRAPPED TO PREVIOUS LOCATION
000140  000142          ,+2             ;TRAP ENTRANCE
000142  000000          HALT            ;TRAPPED TO PREVIOUS LOCATION
000144  000146          ,+2             ;TRAP ENTRANCE
000146  000000          HALT            ;TRAPPED TO PREVIOUS LOCATION
000150  000152          ,+2             ;TRAP ENTRANCE
000152  000000          HALT            ;TRAPPED TO PREVIOUS LOCATION
000154  000156          ,+2             ;TRAP ENTRANCE
000156  000000          HALT            ;TRAPPED TO PREVIOUS LOCATION
000160  000162          ,+2             ;TRAP ENTRANCE
000162  000000          HALT            ;TRAPPED TO PREVIOUS LOCATION
000164  000166          ,+2             ;TRAP ENTRANCE
000166  000000          HALT            ;TRAPPED TO PREVIOUS LOCATION
000170  000172          ,+2             ;TRAP ENTRANCE
000172  000000          HALT            ;TRAPPED TO PREVIOUS LOCATION
000174  000176          ,+2             ;TRAP ENTRANCE
000176  000000          HALT            ;TRAPPED TO PREVIOUS LOCATION
000200  000202          ,+2             ;TRAP ENTRANCE
000202  000000          HALT            ;TRAPPED TO PREVIOUS LOCATION
000204  000206          ,+2             ;TRAP ENTRANCE
000206  000000          HALT            ;TRAPPED TO PREVIOUS LOCATION
000210  000212          ,+2             ;TRAP ENTRANCE
000212  000000          HALT            ;TRAPPED TO PREVIOUS LOCATION
000214  000216          ,+2             ;TRAP ENTRANCE
000216  000000          HALT            ;TRAPPED TO PREVIOUS LOCATION
000220  000222          ,+2             ;TRAP ENTRANCE
000222  000000          HALT            ;TRAPPED TO PREVIOUS LOCATION
000224  000226          ,+2             ;TRAP ENTRANCE
000226  000000          HALT            ;TRAPPED TO PREVIOUS LOCATION
000230  000232          ,+2             ;TRAP ENTRANCE
000232  000000          HALT            ;TRAPPED TO PREVIOUS LOCATION
000234  000236          ,+2             ;TRAP ENTRANCE
000236  000000          HALT            ;TRAPPED TO PREVIOUS LOCATION
000240  000242          ,+2             ;TRAP ENTRANCE
000242  000000          HALT            ;TRAPPED TO PREVIOUS LOCATION
000244  000246          ,+2             ;TRAP ENTRANCE
000246  000000          HALT            ;TRAPPED TO PREVIOUS LOCATION
000250  000252          ,+2             ;TRAP ENTRANCE
000252  000000          HALT            ;TRAPPED TO PREVIOUS LOCATION
000254  000256          ,+2             ;TRAP ENTRANCE
000256  000000          HALT            ;TRAPPED TO PREVIOUS LOCATION
000260  000262          ,+2             ;TRAP ENTRANCE
000262  000000          HALT            ;TRAPPED TO PREVIOUS LOCATION
000264  000266          ,+2             ;TRAP ENTRANCE
000266  000000          HALT            ;TRAPPED TO PREVIOUS LOCATION
000270  000272          ,+2             ;TRAP ENTRANCE
000272  000000          HALT            ;TRAPPED TO PREVIOUS LOCATION
000274  000276          ,+2             ;TRAP ENTRANCE
000276  000000          HALT            ;TRAPPED TO PREVIOUS LOCATION
000300  000302          ,+2             ;TRAP ENTRANCE
```

```
000302  000000                          HALT                    ;TRAPPED TO PREVIOUS LOCATION
000304  000306                          ,+2                     ;TRAP ENTRANCE
000306  000000                          HALT                    ;TRAPPED TO PREVIOUS LOCATION
000310  000312                          ,+2                     ;TRAP ENTRANCE
000312  000000                          HALT                    ;TRAPPED TO PREVIOUS LOCATION
000314  000316                          ,+2                     ;TRAP ENTRANCE
000316  000000                          HALT                    ;TRAPPED TO PREVIOUS LOCATION
000320  000322                          ,+2                     ;TRAP ENTRANCE
000322  000000                          HALT                    ;TRAPPED TO PREVIOUS LOCATION
000324  000326                          ,+2                     ;TRAP ENTRANCE
000326  000000                          HALT                    ;TRAPPED TO PREVIOUS LOCATION
000330  000332                          ,+2                     ;TRAP ENTRANCE
000332  000000                          HALT                    ;TRAPPED TO PREVIOUS LOCATION
000334  000336                          ,+2                     ;TRAP ENTRANCE
000336  000000                          HALT                    ;TRAPPED TO PREVIOUS LOCATION
000340  000342                          ,+2                     ;TRAP ENTRANCE
000342  000000                          HALT                    ;TRAPPED TO PREVIOUS LOCATION
000344  000346                          ,+2                     ;TRAP ENTRANCE
000346  000000                          HALT                    ;TRAPPED TO PREVIOUS LOCATION
000350  000352                          ,+2                     ;TRAP ENTRANCE
000352  000000                          HALT                    ;TRAPPED TO PREVIOUS LOCATION
000354  000356                          ,+2                     ;TRAP ENTRANCE
000356  000000                          HALT                    ;TRAPPED TO PREVIOUS LOCATION
000360  000362                          ,+2                     ;TRAP ENTRANCE
000362  000000                          HALT                    ;TRAPPED TO PREVIOUS LOCATION
000364  000366                          ,+2                     ;TRAP ENTRANCE
000366  000000                          HALT                    ;TRAPPED TO PREVIOUS LOCATION
000370  000372                          ,+2                     ;TRAP ENTRANCE
000372  000000                          HALT                    ;TRAPPED TO PREVIOUS LOCATION
000374  000376                          ,+2                     ;TRAP ENTRANCE
000376  000000                          HALT                    ;TRAPPED TO PREVIOUS LOCATION

                                ;LOAD VECTOR AREA
        000024                          ,=24
000024  016532                          PFAIL                   ;POWER FAIL TRAP
000026  000340                          340
        000030                          ,=30
000030  014756                          EMTSRV                  ;EMT CALLS
000032  000340                          340                     ;HIGHEST PRIORITY
        000034                          ,=34
000034  014550                          SCOPEC                  ;USER TRAP
000036  000000                          0


                                ;LOAD STARTING AREA
        000200                          ,=200
000200  000137  000640                  JMP     @#START
        000300                          ,=300
000300  000137  000640                  JMP     @#START
        000310                          ,=310
000310  000137  000616                  JMP     @#RSTRT

                                ;DATA AREA
        000400                          ,=400
```

```
000400  000000          UBUFF:  0                       ;BUFFER FOR USER SP
        000406                  .=.+4                   ;FOR STACK OVERRUN
000406  177560          TRCSR:  177560                  ;TTY READER STATUS REGISTER
000410  177564          TTCSR:  177564                  ;TTY PUNCH STATUS REGISTER
000412  177566          TTDBR:  177566
000414  000064          TTPVC:  64
000416  000066          TTPST:  66
000420  000000          TTSAV:  0
000422  000100          KWLVC:  100
000424  000102          KWLST:  102
000426  177546          LKCSR:  177546
000430  177514          LPOSR:  177514
000432  177516          LPODR:  177516
000434  000200          LPVC:   200
000436  000202          LPST:   202
000440  177470          RFDAE:  177470                  ;DISK ADDRESS AND ERROR
000442  177466          RFDAR:  177466                  ;DISK ADDRESS REGISTER
000444  177462          RFWC:   177462                  ;WORD COUNT REGISTER
000446  177464          RFCAR:  177464                  ;CURRENT ADDRESS REGISTER
000450  177460          RFCSR:  177460                  ;STATUS REGISTER
000452  177461          RFCSRH: 177461                  ;HIGH BYTE ADDRESS OR CSR
000454  000204          RFVC:   204
000456  000206          RFST:   206
000460  177413          RKDAH:  177413          ;HIGH BYTE DISK ADR
000462  177412          RKDAE:  177412                  ;DISK ADDRESS REGISTER
000464  177406          RKWC:   177406          ;WORD COUNT REGISTER
000466  177410          RKBAR:  177410          ;CURRENT ADDRESS REGISTER
000470  177404          RKCSR:  177404                  ;STATUS REGISTER
000472  177405          RKCSRH: 177405                  ;HIGH BYTE OF CSR
000474  000220          RKVC:   220                     ;TRAP VECTOR
000476  000222          RKST:   222

000500  177572          SR0:    177572                  ;KT11-D  REGISTERS
000502  177600          UPDR0:  177600
000504  177602          UPDR1:  177602
000506  177616          UPDR7:  177616
000510  177640          UPAR0:  177640
000512  177642          UPAR1:  177642
000514  177656          UPAR7:  177656
000516  172300          KPDR0:  172300
000520  172302          KPDR1:  172302
000522  172304          KPDR2:  172304
000524  172316          KPDR7:  172316
000526  172340          KPAR0:  172340
000530  172342          KPAR1:  172342
000532  172344          KPAR2:  172344
000534  172356          KPAR7:  172356

000536  177600          IPORTAB:177600
000540  177640                  177640
000542  172300                  172300
000544  172340          IPOREND:172340
000546  000000          SREG1:  0                       ;STORES KT11-D SWITCH REGISTER SETTINGS
000550  000000          SREG2:  0                       ;STORES SR SETTINGS
000552  177342          TCCM:   177342                  ;CONTROL AND FUNCTION
```

```
000554  177340                  TCST:   177340                  ;GENERAL STATUS
000556  177350                  TCDT:   177350                  ;DATA
000560  177344                  TCWC:   177344                  ;WORD COUNT
000562  177346                  TCBA:   177346                  ;BUS ADDRESS
000564  000214                  TCIV:   214                     ;DECTAPE INTERRUPT VECTOR
000566  000216                  TCSTA:  216
000570  000000                  CURBNK: 0                       ;SAF TO POINT TO CURRENT BANK
000572  000000                  OLDBNK: 0
000574  000000                  CURPAR: 0                       ;ADDRESS OF CURRENT ISAR
000576  000000                  CURPDR:0
000600  000000                  BNKSTR: 0                       ; PC TO POINT TO BEGIN THRU CURRENT SEGMENT
000602  000000                  TRPB:   0
                                ;THE NEXT TWO WORDS ARE THE MEMORY MAP, THE FIRST WORD REPRESENTS
                                ;0-64K WITH ONE BIT REPRESENTING A 4K CONTIGUOUS BLOCK, IF THE
                                ;BIT=1 THAT 4K BLOCK IS PRESENT, THE LSB REPRESENTS 0-4K, THE NEXT
                                ;SIGNIFICANT BIT REPRESENTS 4-8K ANS SO ON,
000604  177777                  MEM0:   177777                  ;0-64K
000606  077777                  MEM1:   77777                   ;64-124K
000610  000001                  COREPT: 1
000612  000604                  MEMUT:  MEM0
000614  000000                  TBANK:  0

                                ;RESTART ADD USING INITIAL SR SETTINGS
000616  012706  017760          RSTRT:  MOV     #KSTACK,R6
000622  012737  016532  000024          MOV     #PFAIL,@#24
000630  113737  177571  000551          MOVB    @#SR+1,@#SREG2+1
000636  000455                          BR      START1

                                ;START UP FOR MINI MONITOR

000640  012706  017760          START:  MOV     #KSTACK,R6              ;SET UP STACK
000644  012737  000137  000200          MOV     #137,@#200              ;RESTORE 200 IF START AT 300
000652  012737  000640  000202          MOV     #START,@#202
000660  005737  000042                  TST     @#42                    ;CHECK FOR MONITOR OPERATION
000664  001433                          BEQ     STARTX
000666  012737  000002  000006          MOV     #RTI,@#6                ;IS THIS AN 11/40
000674  012700  000003                  MOV     #3,R0
000700  000261                          SEC
000702  005737  177772                  TST     @#177772                ;R0=3 IF 11/45
000706  005600                          SBC     R0                      ;R0=2 IF 11/40
000710  000261                          SEC                             ;R0=1 IF 11/20
000712  105737  177777                  TSTB    @#PSR+1                 ;R0=0 IF 11/05
000716  005600                          SBC     R0
000720  005037  177700                  CLR     @#177700
000724  005037  000006                  CLR     @#6
000730  022700  000002                  CMP     #2,R0
000734  001402                          BEQ     .+6
000736  000167  014606                  JMP     LOGIC
000742  005037  000546                  CLR     @#SREG1                 ;RUN ALL SW DOWN IF MONITOR
000746  005037  000550                  CLR     @#SREG2
000752  000407                          BR      START1
000754  013737  177570  000546  STARTX: MOV     @#SR,@#SREG1            ;STORE KT11-D SWITCHES
000762  000000                          HALT
000764  013737  177570  000550          MOV     @#SR,@#SREG2
```

```
000772  004767  013716          START1: JSR     X7,NRALL
000776  012777  077406  177512          MOV     #77400,@KPDR0
001004  012777  007600  177522          MOV     #7600,@KPAR7    ;MAP PAGE 7 TO EXT BANK
001012  012777  077406  177504          MOV     #77406,@KPDR7
001020  005067  177570                  CLR     TBANK
001024  012767  177777  177552          MOV     #177777,MEM0    ;SET UP CORE MAPS
001032  012767  077777  177546          MOV     #77777,MEM1
001040  012767  000001  177542          MOV     #1,COREPT       ;SET UP 4K POINTER
001046  012767  000604  177536          MOV     #MEM0,MEMUT
001054  012777  077406  177440          MOV     #77406,@KPDR2   ;BEING CHECKED FOR
001062  012737  001132  000004          MOV     #TMEMEX,@#4     ;SET UP FOR TIME OUTS
001070  005037  000006                  CLR     @#6
001074  052777  000001  177376          BIS     #1,@SR0
001102  016777  177500  177422  MAP1:   MOV     TBANK,@KPAR2    ;MAP KERNEL PAGE 2 TO BANK
001110  005737  041000                  TST     @#41000         ;1ST K PRESENT
001114  005737  045000                  TST     @#45000         ;2ND K PRESENT
001120  005737  051000                  TST     @#51000         ;3RD K PRESENT
001124  005737  055000                  TST     @#55000         ;4TH K PRESENT
001130  000404                          BR      MOVEPT          ;OK, FULL 4K BLOCK PRESENT
001132  046777  177452  177452  TMEMEX: BIC     COREPT,@MEMUT   ;NO, BLOCK NOT PRESENT
001140  022626                          CMP     (SP)+,(SP)+     ;ADJUST STACK POINTER
001142  062767  000200  177444  MOVEPT: ADD     #200,TBANK      ;UPDATE BANK POINTER
001150  006367  177434                  ASL     COREPT
001154  103006                          BCC     MAP2            ;THIS 1ST MEM WORD DONE
001156  012767  000001  177424          MOV     #1,COREPT
001164  012767  000606  177420          MOV     #MEM1,MEMUT
001172  022767  007600  177414  MAP2:   CMP     #7600,TBANK     ;EXTERNAL BANK YET
001200  001340                          BNE     MAP1            ;NO,NOT YET?
001202  012767  000001  177400          MOV     #1,COREPT       ;RE-INIT
001210  012767  000604  177374          MOV     #MEM0,MEMUT
001216  042777  000001  177254          BIC     #1,@SR0
001224  012767  000001  013410          MOV     #1,ICOUNT
001232  004767  015242                  JSR     X7,CRLF
001236  012737  014550  000034          MOV     #SCOPEC,@#34
001244  005037  000036                  CLR     @#36            ;INITIALIZE SCOPE CALL TO KERNEL STATUS
001250  012737  014756  000030          MOV     #EMTSRV,@#30
001256  012737  000340  000032          MOV     #340,@#32
001264  012737  005376  014646          MOV     #BEGIN,@#RETURN
001272  005037  014644                  CLR     @#SCOPEF
001276  012737  000340  177776          MOV     #340,@#PSR      ;LOCK OUT INTERRUPTS
001304  005037  016260                  CLR     @#PRTON         ;PRINT ROUTINE BUSY FLAG
001310  000005                          RESET
001312  012737  002260  000004          MOV     #NODEV,@#4              ;RETURN FOR NO DEVICE
001320  005037  000006                  CLR     @#6
001324  005067  001362                  CLR     DATA2           ;BASE DATA FOR TTY TELEPRINTER
001330  033727  000550  000001          BIT     @#SREG2,#1              ;INHIBIT TTY OUTPUT?
001336  001006                          BNE     ST3             ;YES, GO CHECK NEXT,
001340  012777  002724  177046          MOV     #TYOUTR,@TTPVC  ;NO, SETUP INTERRUPT VECTOR
001346  052777  000100  177034          BIS     #100,@TTCSR     ;START TTY OUTPUT
001354  032737  000010  000550  ST3:    BIT     #10,@#SREG2     ;INHIBIT RK DISK
001362  001026                          BNE     ST4             ;YES, SKIP OVER
001364  005777  177100                  TST     @RKCSR          ;PRESENT
001370  012777  003322  177076          MOV     #IRK,@RKVC      ;SETUP VECTOR RETURNS
001376  012777  000240  177072          MOV     #240,@RKST
```

```
001404  012767  043503  001752          MOV     #43503,RKFUNCT
001412  005077  177044                  CLR     @RKDAE                  ;INIT
001416  016777  002102  177042          MOV     LLIMIT,@RKBAR           ;CORE BASE
001424  016777  002076  177032          MOV     WORDCT,@RKWC            ;TRANSFER LENGTH
001432  116777  001726  177030          MOVB    RKFUNCT,@RKCSR
001440  033727  000550  000020  ST4:    BIT     @#SREG2,#20             ;INHIBIT LINE CLOCK?
001446  001015                          BNE     ST5                     ;YES, GO CK NEXT
001450  005777  176752                  TST     @LKCSR                  ;PRESENT
001454  012777  003002  176740          MOV     #LK3,@KWLVC
001462  012777  000300  176734          MOV     #300,@KWLST
001470  005067  001402                  CLR     TIME                    ;NO, INITIALIZE COUNT
001474  052777  000100  176724          BIS     #100,@LKCSR             ;START LINE CLOCK
001502  033727  000550  000040  ST5:    BIT     @#SREG2,#40             ;TEST FOR INHIBITING RF11 DISK
001510  001026                          BNE     ST6                     ;SKIP IF SET
001512  005777  176732                  TST     @RFCSR                  ;PRESENT?
001516  012777  003416  176730          MOV     #IRF,@RFVC              ;SET UP TRAP RETURN
001524  012777  000240  176724          MOV     #240,@RFST
001532  012767  043503  001762          MOV     #43503,RFFUNCT          ;WRITE CHECK/WRITE
001540  105277  176706                  INCB    @RFCSRH                 ;INITIALIZE DISK=DAR,DAE
001544  016777  001756  176672          MOV     WORDCT,@RFWC            ;LENGTH OF TRANSFER
001552  016777  001746  176666          MOV     LLIMIT,@RFCAR           ;CORE ADDRESS OF START OF TRANSFER
001560  116777  001736  176662          MOVB    RFFUNCT,@RFCSR          ;START RF11 READ OR WRITE
001566  033727  000550  000100  ST6:    BIT     @#SREG2,#100            ;CHECK FOR INHIBITING TC11 DECTAPE
001574  001013                          BNE     ST7                     ;SKIP IF SET
001576  005777  176752                  TST     @TCST                   ;PRESENT?
001602  012777  003536  176754          MOV     #FEND2,@TCIV            ;GO TO END ZONE ON INTERRUPT
001610  012777  000300  176750          MOV     #300,@TCSTA
001616  012777  004503  176726          MOV     #R+IE+RB+D8,@TCCM       ;START REVERSE READ BLOCK NUMBER
001624  033727  000550  000200  ST7:    BIT     @#SREG2,#200            ;INHIBIT LINE PRINTER?
001632  001032                          BNE     ST8                     ;YES, GO CK NEXT
001634  005777  176570                  TST     @LPCSR                  ;PRESENT?
001640  012737  001720  000004          MOV     #ST8,@#4                ;DON'T CHANGE 200 IF NO SUCH DEVICE
001646  012767  000137  001226          MOV     #137,SOLPAT             ;RESET FOR START OF LINE PATTERN
001654  012767  000117  001312          MOV     #79.,CLINCT             ;LINE COUNT
001662  012767  000137  001214          MOV     #137,CURPAT
001670  012777  000014  176534          MOV     #14,@LPDBR              ;LINE FEED TO POSITION BUFFER
001676  012777  003124  176530          MOV     #LPINTR,@LPVC           ;INTERRUPT ENABLE
001704  012777  000200  176524          MOV     #200,@LPST              ;PROCESSOR LEVEL 4
001712  012777  000100  176510          MOV     #100,@LPCSR             ;INTERRUPT ENABLE
001720  005037  000602          ST8:    CLR     @#TRPB                  ;NO "T" BIT FIRST PASS
001724  005037  000006                  CLR     @#6                     ;CHANGE ADDRESS ERROR VECTOR TO CAUSE
001730  012737  000006  000004          MOV     #6,@#4                  ;HALT ON A TRAP TO 4
001736  004737  016574                  JSR     %7,@#USER               ;FOR USER I/O PROGRAM INSERTION
001742  004767  000326                  JSR     %7,DET1                 ;CHECK FOR CORE EXPANSION
001746  032737  000001  000546          BIT     #1,@#SREG1              ;INHIBIT KT11=D?
001754  001106                          BNE     MODE                    ;YES = GO SETUP USER
001756  004767  012732                  JSR     %7,NRALL                ;NO = MAKE ALL SEGMENTS INITIALLY NON=RESIDENT
001762  012777  077406  176534          MOV     #77406,@KPDR7
001770  012777  007600  176536          MOV     #7600,@KPAR7
001776  032737  000006  000546          BIT     #6,@#SREG1              ;INHIBIT USER/KERNEL OR 4K AS 32K?
002004  001415                          BEQ     SEGM1                   ;NO = BRANCH
002006  012701  000007                  MOV     #7,R1                   ;YES = MAP KERNEL ASR'S 2=6 TO PA
002012  016702  176510                  MOV     KPAR0,R2
002016  005003                          CLR     R3
```

```
002020  010312                  SETEX:  MOV     R3,@R2
002022  012762  077406  177740          MOV     #77406,-40(R2)
002030  005722                          TST     (R2)+
002032  062703  000200                  ADD     #200,R3
002036  077110                          SOB     R1,SETEX
002040  012777  077406  176450  SEGM1:  MOV     #77406,@KPDR0           ;MAP KERNEL 0 TO BANK 0, RW
002046  032737  000004  000546          BIT     #4,@#SREG1              ;INHIBIT RUNNING 4K AS 32K?
002054  001416                          BEQ     USEALL                  ;NO, SETUP FOR RUNNING 4K AS 32K
002056  012701  000010                  MOV     #10,R1                  ;YES, MAP ALL USER ASR'S TO PA
002062  016702  176422                  MOV     UPAR0,R2
002066  005003                          CLR     R3
002070  010312                  SETUSE: MOV     R3,(R2)
002072  062703  000200                  ADD     #200,R3
002076  012762  077406  177740          MOV     #77406,-40(R2)
002104  005722                          TST     (R2)+
002106  077110                          SOB     R1,SETUSE
002110  000425                          BR      SETSEG
002112  012777  077406  176362  USEALL: MOV     #77406,@UPDR0           ;MAP USER ASR0 TO BANK 0, RW
002120  012737  000000  000570          MOV     #0,@#CURBNK             ;CURRENT SAR CONTENTS
002126  012767  000001  176454          MOV     #1,COREPT               ;INIT MAP POINTERS
002134  012767  000004  176450          MOV     #MEM0,MEMU?
002142  016767  176342  176424          MOV     UPAR0,CURPAR            ;CURRENT SEGMENT REGISTER ADDRESSES
002150  016767  176326  176420          MOV     UPDR0,CURPDR
002156  012767  005376  176414          MOV     #BEGIN,BNKSTR           ;CURRENT STARTING PC
002164  052777  000001  176306  SETSEG: BIS     #1,@SR0                 ;SET KT11-D BIT
002172  042737  000340  177776  MODE:   BIC     #340,@#PSR              ;PRIORITY LEVEL 0
002200  032737  000002  000546          BIT     #2,@#SREG1              ;INHIBIT USER/KERNEL?
002206  001016                          BNE     MAIN+2                  ;YES - SKIP OVER
002210  052737  140000  000036          BIS     #140000,@#36            ;SET USER BIT IN SCOPE STATUS
002216  012746  000400                  MOV     #UBUFF,-(R6)
002222  052737  030000  177776          BIS     #30000,@#PSR
002230  006606                          MTPI    SP                      ;SET UP USER STACK
002232  012737  140000  177776          MOV     #140000,@#PSR           ;CHANGE TO USER
002240  000401                          BR      .+4
002242  000001                  MAIN:   WAIT
002244  033727  000550  002000          BIT     @#SREG2,#2000           ;INHIBIT PROCESSOR TEST
002252  001373                          BNE     MAIN
002254  000167  003116                  JMP     BEGIN

                                ;NON-EXISTING DEVICE SERVICE
002260  162716  000006          NODEV:  SUB     #6,(SP)                 ;ALTER PC RETURN
002264  042766  000017  000002          BIC     #17,2(SP)               ;CLEAR Z BIT ON STACK
002272  000002                          RTI

                                ;PDP-11 MEMORY DETERMINATION AND SETUP?
                                ;USE WITH VARIABLE CORE QUANTITY SYSTEMS/
002274  012767  104010  012204  DET1:   MOV     #EOB,DONE               ;RESTORE INITIAL CODE
002302  032767  000007  176236          BIT     #7,SREG1                ;INHIBIT RUNNING 4K AS 32K USER?
                                                                        ;OR INHIBIT SEGMENTATION?
002310  001001                          BNE     .+4                     ;YES - ALLOW CORE EXPANSION
002312  000207                          RTS     %7                      ;NO - INHIBIT CORE EXPANSION
002314  032737  000040  000546          BIT     #40,@#SREG1             ;CHECK VARIABLE CORE SWITCH
002322  001401                          BEQ     DET4                    ;USE VARIABLE CORE ROUTINE
002324  000207                          RTS     %7                      ;4K ONLY (SWITCH SET)
```

```
002326  012737  002376  000004  DET4:   MOV     #DET2,@#4               ;TRAP VECTOR SETUP
002334  012737  000340  000006          MOV     #340,@#6                ;TRAP STATUS SETUP
002342  000241                          CLC
002344  005537  037770          EIGHT:  ADC     @#37770                 ;CHECK FOR 8K
002350  005537  057770                  ADC     @#57770                 ;CHECK FOR 12K
002354  005537  077770                  ADC     @#077770                ;CHECK FOR 16K
002360  005537  117770                  ADC     @#117770                ;CHECK FOR 20K
002364  005537  137770                  ADC     @#137770                ;CHECK FOR 24K
002370  005537  157770                  ADC     @#157770                ;CHECK FOR 28K
002374  000437                          BR      STRT28
002376  012602                  DET2:   MOV     (6)+,X2                 ;RETRIEVE TRAP PC
002400  005726                          TST     (6)+                    ;DISCARD TRAP STATUS WORD
002402  062702  000060                  ADD     #STRT4-EIGHT-4,R2
002406  000112                          JMP     @R2

002410  005000                  MOVE:   CLR     X0                      ;SET UP MAIN CORE POINTER
002412  010102                          MOV     X1,X2
002414  062702  014714                  ADD     #D+2,X2                 ;SET UP MAX CORE MOVE
002420  012021                          MOV     (0)+,(1)+               ;MOVE WORD
002422  020201                          CMP     X2,X1                   ;MOVE COMPLETE?
002424  001375                          BNE     .-4                     ;MOVE ANOTHER WORD
002426  000207                          RTS     X7                      ;MOVE COMPLETE
002430  000521                  STRT4:  BR      DET3
002432  000240                          NOP
002434  000240                          NOP
002436  004767  000110                  JSR     X7,XFER8                ;START 8K TRANSFER
002442  000506                          BR      MOD4                    ;START 4K MODIFY
002444  004767  000072                  JSR     X7,XFER12               ;START 12K TRANSFER
002450  000475                          BR      MOD8                    ;START 8K MODIFY
002452  004767  000054                  JSR     X7,XFER16               ;START 16K TRANSFER
002456  000464                          BR      MOD12                   ;START 12K MODIFY
002460  004767  000036                  JSR     X7,XFER20               ;START 20K TRANSFER
002464  000453                          BR      MOD16                   ;START 16K MODIFY
002466  004767  000020                  JSR     X7,XFER24               ;START 24K TRANSFER
002472  000442                          BR      MOD20                   ;START 20K MODIFY
002474  004767  000002          STRT28: JSR     X7,XFER28               ;START 28K TRANSFER
002500  000431                          BR      MOD24                   ;START 24K MODIFY
002502  012701  140000          XFER28: MOV     #140000,X1              ;SET UP MOVE START LOCATION
002506  004767  177676                  JSR     X7,MOVE                 ;GO TO MOVE SUBROUTINE
002512  012701  120000          XFER24: MOV     #120000,X1
002516  004767  177666                  JSR     X7,MOVE
002522  012701  100000          XFER20: MOV     #100000,X1
002526  004767  177656                  JSR     X7,MOVE
002532  012701  060000          XFER16: MOV     #60000,X1
002536  004767  177646                  JSR     X7,MOVE
002542  012701  040000          XFER12: MOV     #40000,X1
002546  004767  177636                  JSR     X7,MOVE
002552  012701  020000          XFER8:  MOV     #20000,X1
002556  004767  177626                  JSR     X7,MOVE
002562  000207                          RTS     X7                      ;RETURN FROM TRANSFERS
002564  012767  000137  131714  MOD24:  MOV     #137,DONE+120000
002572  012767  145344  131710          MOV     #BEGINX+140000,DONE+120002
002600  012767  000137  111700  MOD20:  MOV     #137,DONE+100000
002606  012767  125344  111674          MOV     #BEGINX+120000,DONE+100002
```

```
002614  012767  000137  071664  MOD16:  MOV     #137,DONE+60000
002622  012767  105344  071660          MOV     #BEGINX+100000,DONE+60002
002630  012767  000137  051650  MOD12:  MOV     #137,DONE+40000
002636  012767  065344  051644          MOV     #BEGINX+60000,DONE+40002
002644  012767  000137  031634  MOD8:   MOV     #137,DONE+20000
002652  012767  045344  031630          MOV     #BEGINX+40000,DONE+20002
002660  012767  000137  011620  MOD4:   MOV     #137,DONE
002666  012767  025344  011614          MOV     #BEGINX+20000,DONE+2
002674  005037  000006          DE%3:   CLR     @#6
002700  012737  000006  000004          MOV     #6,@#4
002706  000207                          RTS     %7

                                ;TTY TRANSMITTER PRINT VALUES 0 TO 377?
002710  005027  000000          TYOUT:  CLR     #0                      ;INITAL DATA
        000712                          DATA2=.-2
002714  016777  177772  175470  TYOUT1: MOV     DATA2,@TTOBR            ;OUTPUT TO DEVICE
002722  000002                          RTI                             ;RETURN TO MAINLINE**
002724  017767  175460  175466  TYOUTR: MOV     @TTCSR,TTSAV
002732  105767  175462                  TSTB    TTSAV                   ;TEST FOR DONE
002736  100401                          BMI     .+4                     ;BRANCH IF FLAG FOUND
002740  104006                          HLT                             ;FALSE INTERRUPT RETURN
002742  005267  177744                  INC     DATA2                   ;INCREMENT DATA
002746  022767  000400  177736          CMP     #400,DATA2              ;TEST DATA FOR UPPER LIMIT
002754  001755                          BEQ     TYOUT                   ;AT UPPER LIMIT START OVER
002756  000756                          BR      TYOUT1                  ;FINISH REST OF DATA

                                ;TEST OF LINE CLOCK, INTERRUPT FOR 55 SECONDS THEN STALL FOR 5 SECONDS.
002760  005037  003076          LK1:    CLR     @#TIME                  ;CLEAR LINE CLOCK TIMER
002764  052777  000100  175434          BIS     #100,@LKCSR
002772  052737  000100  177776          BIS     #100,@#PSR
003000  000002          LK2:    RTI
003002  105777  175420          LK3:    TSTB    @LKCSR
003006  100401                          BMI     .+4
003010  104006                          HLT                             ;FALSE INTERRUPT
003012  042777  000200  175406          BIC     #200,@LKCSR
003020  005237  003076          LK4:    INC     @#TIME                  ;HERE ON INTERRUPTS
003024  022737  006344  003076          CMP     #3300,.,@#TIME          ;55 SEC YET?
003032  103362                          BHIS    LK2                     ;BR IF NOT
003034  042777  000100  175364          BIC     #100,@LKCSR
003042  042737  000100  177776          BIC     #100,@#PSR              ;LOWER PRIORITY
003050  022737  007020  003076          CMP     #3600,.,@#TIME          ;OME MINUTE YET
003056  001740                          BEQ     LK1                     ;YES RESET TIMER
003060  105777  175342                  TSTB    @LKCSR                  ;NO, SKIP TILL MINUTE UP
003064  100375                          BPL     .-4
003066  042777  000200  175332          BIC     #200,@LKCSR             ;CLEAR FLAG
003074  000751                          BR      LK4
003076  000000          TIME:   0

                                ;LINE PRINTER SHOULD RAISE PROCESSOR PRIORITY TO LEVEL OF LINE PRINTER/
                                ;INTERRUPT VECTOR IS 200/
003100  012727  000000  000000  LP1:    MOV     #0,#0                   ;START OF LINE TO CURRENT
        003104                          CURPAT=.-2                      ;CHARACTER BEING PRINTED
        003102                          SOLPAT=.-4                      ;START OF LINE CHARACTER
003106  016777  177772  175316  LP2:    MOV     CURPAT,@LPDBR           ;CURRENT PATTERN TO LINE PRINTER
```

```
003114  105777  175310                  TSTB    @LPCSR
003120  100420                          BMI     LP6
003122  000002                          RTI                             ;RETURN TO MAIN LINE
003124  105777  175300          LPINTR: TSTB    @LPCSR                  ;TEST FOR FLAG
003130  100414                          BMI     LP6
003132  005737  000042                  TST     @#42                    ;MONITOR LOAD
003136  001410                          BEQ     LP7                     ;NO, ERROR
003140  032777  100000  175262          BIT     #100000,@LPCSR          ;YES, IS ERROR SET
003146  001404                          BEQ     LP7                     ;NO, ERROR
003150  042777  000100  175252          BIC     #100,@LPCSR             ;DIS ABLE INTERRUPT
003156  000002                          RTI
003160  104006          LP7:    HLT                                     ;FALSE RETURN FROM MAIN LINE
003162  026727  000006  000117  LP6:    CMP     CLINCT,#79.             ;TEST FOR END OF LINE
003170  001415                          BEQ     LP4                     ;GO GENERATE CR/LF
003172  005227  000000                  INC     #0                      ;INCREMENT LINE POSITION COUNT
        003174                          CLINCT+.-2                      ;POSITION OF LINE
003176  026727  177702  000137          CMP     CURPAT,#137             ;TEST FOR MAXIMUM PATTERN
003204  001403                          BEQ     LP3                     ;YES - GO TO LP3 AND RESET
003206  005267  177672                  INC     CURPAT                  ;NO - INCREMENT TO NEXT PATTERN
003212  000735                          BR      LP2                     ;GO SEND IT TO LINE PRINTER
003214  012767  000040  177662  LP3:    MOV     #40,CURPAT              ;RESET PATTERN AND SEND TO PRINTER
003222  000731                          BR      LP2                     ;SENT TO LINE PRINTER
003224  005067  177744          LP4:    CLR     CLINCT                  ;RESET LINE COUNT
003230  012777  000012  175174          MOV     #12,@LPDBR              ;LINE FEED
003236  105777  175166                  TSTB    @LPCSR
003242  100375                          BPL     .-4
003244  026727  177632  000137          CMP     SOLPAT,#137             ;START OF LINE PATTERN
003252  001403                          BEQ     LP5
003254  005267  177622                  INC     SOLPAT                  ;INCREMENT START OF LINE
003260  000707                          BR      LP1
003262  012767  000040  177612  LP5:    MOV     #40,SOLPAT              ;RESET START OF LINE
003270  000703                          BR      LP1                     ;PRINT

                                ;RK11 DISK TEST INTERRUPT LEVEL 5, 2000 WORD TRANSFERS
003272  005077  175164          RKSTART:        CLR     @RKDAE  ;INIT
003276  013777  003524  175162  RK1:    MOV     @#LLIMIT,@RKBAR         ;CORE BASE
003304  013777  003526  175152          MOV     @#WORDCT,@RKWC          ;TRANSFER LENGTH
003312  113777  003364  175150          MOVB    @#RKFUNCT,@RKCSR        ;WRITE OR WRITE CK TO DSK
003320  000002                          RTI                             ;RETURN TO MAINLINE
003322  032777  100200  175140  IRK:    BIT     #100200,@RKCSR          ;INTERRUPT RETURN
003330  003002                          BGT     .+6
003332  104006                          HLT
003334  000756                          BR      RKSTART
003336  032777  000037  175116          BIT     #37,@RKDAE              ;DISK AT UPPER LIMIT?
003344  001354                          BNE     RK1
003346  122777  000031  175104          CMPB    #31,@RKDAH
003354  001350                          BNE     RK1
003356  000337  003364                  SWAB    @#RKFUNCT               ;CHANGE COMMAND
003362  000743                          BR      RKSTART                 ;RESTART NEW TRANSFER OF DISK
003364  000000          RKFUNCT:        0

                                ;RF11 DISK
003366  105277  175060          RFSTART:        INCB    @RFCSRH         ;INITIALIZE DISK - DAR-DAE
003372  013777  003524  175046  RF1:    MOV     @#LLIMIT,@RFCAR         ;CORE BASE
```

```
003400  013777  003526  175036          MOV     @#WORDCT,@RFWC          ;LENGTH OF TRANSFER
003406  113777  003522  175034          MOVB    @#RFFUNCT,@RFCSR
                                                                        ;WRITE OR WRITE CHECK TO DISK
003414  000002                          RTI                             ;RETURN TO MAINLINE CODE
003416  105777  175026          IRF1:   TSTB    @RFCSR                  ;INTERRUPT VECTOR POINTS HERE
003422  100402                          BMI     .+6
003424  104006                          HLT                             ;RF11 READY NOT UP
003426  000757                          BR      RFSTART
003430  005777  175014                  TST     @RFCSR                  ;ERROR SET?
003434  100012                          BPL     ERROK                   ;BRANCH IF NOT
003436  032777  020000  175004          BIT     #20000,@RFCSR           ;YES-WRITE CHECK ERROR?
003444  001404                          BEQ     ERRSET                  ;NO-BRANCH
003446  104006                          HLT                             ;YES-RF11 WRITE CHECK ERROR
003450  000337  003522                  SWAB    @#RFFUNCT               ;CHANGE COMMAND TO DO WRITE
003454  000744                          BR      RFSTART
003456  104006          ERRSET: HLT                                     ;RF11 ERROR SET-NOT WRITE CHECK
003460  000742                          BR      RFSTART
003462  005777  174756          ERROK:  TST     @RFWC
003466  100002                          BPL     .+6
003470  104006                          HLT                             ;RF-11 WORD COUNT NOT ZERO
003472  000735                          BR      RFSTART
003474  122777  000003  174736          CMPB    #3,@RFDAE               ;DISK AT UPPER LIMIT? 7-2, 17-4, 37-8
003502  001333                          BNE     RF1                     ;NO
003504  027727  174732  174000          CMP     @RFDAR,#174000          ;AS FAR ON DISK AS WE CAN GO
003512  101727                          BLOS    RF1                     ;NO
003514  000337  003522                  SWAB    @#RFFUNCT               ;CHANGE COMMAND
003520  000722                          BR      RFSTART                 ;RESTART NEW TRANSFER OF DISK
003522  000000          RFFUNCT:        0                               ;DISK COMMAND
003524  005376          LLIMIT: BEGIN                                   ;FIRST CORE ADDRESS OF TRANSFER
003526  176000          WORDCT: -2000                                   ;LENGTH OF TRANSFER

                        ;DECTAPE DIAGNOSTIC ROUTINE.  THE TAPE IS FIRST DRIVEN TO THE FORWARD
                        ;END ZONE, THE DESIRED DATA IS THEN GENERATED IN THE DECTAPE BUFFER AREA
                        ;AND DATA IS WRITTEN ONTO ALL BLOCKS FROM THE BLOCK NUMBER IN TCFRST
                        ;THRU THE BLOCK NUMBER IN TCLAST.  BLOCK NUMBERS ARE ALSO CHECKED FOR
                        ;BEING IN ORDER, AFTER THE BLOCK NUMBER IN TCLAST IS WRITTEN, TAPE IS
                        ;DRIVEN INTO THE REVERSE END ZONE.
                        ;THE TAPE IS THEN STARTED IN REVERSE, AND WHEN THE CLOSEST BLOCK THAT
                        ;WAS WRITTEN (TCLAST) IS FOUND, IT IS READ INTO THE DECTAPE BUFFER AREA.
                        ;THE PROGRAM INTERRUPT REQUEST FACILITY IS THEN USED TO BOOK A REQUEST
                        ;FOR CHECKING THE DATA AT LEVEL 3, AND NO FURTHER DATA IS READ IN
                        ;UNTIL THAT DATA HAS BEEN CHECKED.  AFTER IT IS CHECKED, THE DATA IS
                        ;SCRAMBLED TO GUARANTEE THAT NEW DATA IS REALLY READ IN NEXT TIME. WHILE
                        ;THIS IS GOING ON, BLOCK NUMBERS ARE CHECKED FOR BEING IN ORDER AS THE
                        ;TAPE TRAVELS TOWARD THE FORWARD END ZONE.ONCE THE DATA IS FULLY CHECKED
                        ;THE NEXT BLOCK THAT COMES UP IS READ IN AND THE PROCESS REPEATED.  ONCE
                        ;THE BLOCK WHOSE NUMBER IS IN TCFRST HAS BEEN READ, THE TAPE IS DRIVEN
                        ;INTO THE FORWARD END ZONE AND THE WHOLE SEQUENCE IS REPEATED.

                        ;FUNCTION VALUES IN CSR
                        ;D711 DEC TAPE
000004                  RD=4                            ;READ DATA
000014                  WD=14                           ;WRITE DATA
000002                  RB=2
000500                  IE=500                          ;INTERRUPT ENABLE-UNIT 1
```

```
                000001                  DO=1                            ;DO = THE FUNCTION
                004000                  R=4000                          ;REVERSE

003530          000000                  TCFIRST: 0                      ;FIRST BLOCK TO BE SEARCHED FOR
003532          001101                  TCLAST: 577.                    ;LAST BLOCK TO BE SEARCHED FOR
003534          000000                  TCEXPE: 0                       ;THE BLOCK THAT IS EXPECTED

                                        ;GO TO FORWARD END ZONE
003536  012777  003536  175020  FEND2:  MOV     #FEND2,@TCIV            ;END ZONE VECTOR SETUP
003544  005777  175004                  TST     @TCST                   ;TEST FOR END ZONE
003550  100403                          BMI     FEND1                   ;AT END ZONE?
003552  105277  174774                  INCB    @TCCM                   ;SET DO = NO DELAY
003556  000002                          RTI                             ;NO = WAIT SOME MORE
003560  012777  003610  174776  FEND1:  MOV     #TCF1,@TCIV             ;YES = NEW VECTOR
003566  042777  104000  174756          BIC     #104000,@TCCM           ;SEARCH BLOCK FOWARD
003574  016767  177730  177732          MOV     TCFIRST,TCEXPE          ;COUNT WHEN THIS BLOCK IS FOUND
003602  105277  174744          TCF1A:  INCB    @TCCM                   ;SET DO
003606  000002                          RTI                             ;RETURN ON NEXT BLOCK
003610  032777  100200  174734  TCF1:   BIT     #100200,@TCCM           ;ANY ERROR ON READ?
003616  100001                          BPL     .+4
003620  104006                          HLT                             ;TC ERROR SET = FORWARD READ BLOCK
003622  001001                          BNE     .+4                     ;DONE FLAG UP?
003624  104006                          HLT                             ;FALSE INTERRUPT
003626  027767  174724  177700          CMP     @TCDT,TCEXPE            ;IS THIS OUR BLOCK FOR SYNC
003634  002762                          BLT     TCF1A                   ;NO=READ SOME MORE BLOCKS
003636  001401                          BEQ     TCF2                    ;YES
003640  104006                          HLT                             ;WE PASSED THE BLOCK

003642  012777  003656  174714  TCF2:   MOV     #TCF3,@TCIV             ;VECTOR FOR SEQUENTIAL READS
003650  105277  174676                  INCB    @TCCM                   ;SET DO
003654  000002                          RTI                             ;RETURN AND TEST SEQENTIAL BLOCKS

                                        ;FIND SEQUENTIAL BLOCK AT FOWARD DIRECTION
003656  032777  100200  174666  TCF3:   BIT     #100200,@TCCM           ;TEST ERROR AND READY
003664  100001                          BPL     .+4
003666  104006                          HLT                             ;FOWARD READ ERROR TC=11
003670  001001                          BNE     .+4
003672  104006                          HLT                             ;FALSE INTERRUPT ON TC=11
003674  027767  174656  177630          CMP     @TCDT,TCLAST            ;HAVE WE TESTED ALL BLOCKS
003702  001414                          BEQ     REND2                   ;YES DRIVE UNIT IN END ZONE TO START OVER
003704  005267  177624                  INC     TCEXPE                  ;NO=INCREMENT EXPECTED COUNT
003710  027767  174642  177616          CMP     @TCDT,TCEXPE            ;IS CURRENT BLOCK CORRECT
003716  001401                          BEQ     .+4
003720  104006                          HLT                             ;FAILED IN FOWARD READ TO FIND NEXT BLOCK
003722  000427                          BR      TCWBK                   ;THIS ROUTINE WRITES A BLOCK
003724  105277  174622          TCF4:   INCB    @TCCM                   ;SET DO
003730  000002                          RTI
003732  000701                  XFEND2: BR      FEND2                   ;INDIRECT LINK

                                        ;MOVE TAPE TO REVERSE END ZONE
003734  012777  003734  174622  REND2:  MOV     #REND2,@TCIV            ;END ZONE VECTOR SETUP
003742  016767  177564  177564          MOV     TCLAST,TCEXPE           ;SET UP FOR REVERSE SEARCH
003750  005777  174600                  TST     @TCST                   ;IN END ZONE
003754  100403                          BMI     REND1                   ;YES = START TO TURN UNIT AROUND
```

```
003756  105277  174570                  INCB    @TCCM           ;SET DO
003762  000002                          RTI                     ;NO - WAIT TILL WE ARE
003764  012777  004503  174560  REND1:  MOV     #R-IE-RB-DO,@TCCM ;FUNCTION = READ BLOCK, REVERSE AND GO
003772  012777  004062  174564          MOV     #TCR1,@TCIV     ;SET UP NEW INTERRUPT VECTOR
004000  000002                          RTI
                                ;WRITE FORWARD ALL BLOCKS EXCEPT 0

004002  012777  004034  174554  TCWBK:  MOV     #TCWB1,@TCIV    ;INTERRUPT VECTOR FOR WRITE
004010  012777  177400  174542          MOV     #-400,@TCWC     ;ONE BLOCK
004016  012777  004344  174536          MOV     #TCWBUF,@TCBA   ;THE WRITE BUFFER ADDRESS
004024  112777  000515  174520          MOVB    #IE-WD-DO,@TCCM ;WRITE THE BLOCK
004032  000002                          RTI                     ;RETURN WHEN BLOCK IS WRITTEN
004034  005777  174512          TCWB1:  TST     @TCCM           ;ANY ERRORS
004040  100001                          BPL     .+4
004042  104006                          HLT
004044  012777  003656  174512          MOV     #TCF3,@TCIV     ;SEARCH BLOCK VECTOR
004052  112777  000502  174472          MOVB    #IE-RB,@TCCM    ;READ BLOCK
004060  000721                          BR      TCF4            ;FIND THE NEXT BLOCK

004062  032777  100200  174462  TCR1:   BIT     #100200,@TCCM   ;TEST FOR ERROR AND READY
004070  100001                          BPL     .+4
004072  104006                          HLT                     ;DECTAPE ERROR ON READ BLOCK REVERSE
004074  001001                          BNE     .+4
004076  104006                          HLT                     ;FALSE INTERRUPT FROM DECTAPE
004100  027767  174452  177426          CMP     @TCDT,TCEXPE    ;IS IT OUR FIRST BLOCK
004106  001406                          BEQ     TCR2            ;YES - GO TEST THE REST
004110  002002                          BGE     TCR1A           ;NO - HAVE WE PASSED THE BLOCK
004112  104006                          HLT                     ;WE PASS OUR BLOCK
004114  000707                          BR      REND2           ;GO TO END ZONE AND TRY AGAIN
004116  105277  174430          TCR1A:  INCB    @TCCM           ;SET DO
004122  000002                          RTI                     ;WE FOUND OUR FIRST BLOCK
004124  012777  004140  174432  TCR2:   MOV     #TCR3,@TCIV     ;SET UP INTERRUPT TO TEST ALL BLOCKS
004132  105277  174414                  INCB    @TCCM           ;SET DO
004136  000002                          RTI                     ;WAIT FOR NEXT BLOCK TO INTERRUPT

                                ;FIND SEQUENTIAL BLOCK IN REVERSE DIRECTION
004140  032777  100200  174404  TCR3:   BIT     #100200,@TCCM   ;TEST FOR READ AND ERROR
004146  100001                          BPL     .+4
004150  104006                          HLT                     ;ERROR READING SEQUENTIAL BLOCK IN REVERSE
004152  001001                          BNE     .+4
004154  104006                          HLT                     ;FALSE DECTAPE INTERRUPT
004156  026777  177346  174372          CMP     TCFIRST,@TCDT   ;DID WE DO ALL THE BLOCKS
004164  001662                          BEQ     XFEND2          ;YES - GO TO END ZONE TO RESTART
004166  005367  177342                  DEC     TCEXPE          ;NO - DECREMENT BLOCK NUMBER
004172  027767  174360  177334          CMP     @TCDT,TCEXPE    ;TEST SEQUENTAIL BLOCK IN REVERSE
004200  001401                          BEQ     .+4
004202  104006                          HLT                     ;TEST SEQUENTIAL READ BLOCK IN REVERSE FAILED
004204  000403                          BR      TCRBK           ;THIS ROUTINE READ A BLOCK
004206  105277  174340          TCR4:   INCB    @TCCM           ;SET DO
004212  000002                          RTI                     ;LETS TRY A NEW BLOCK

                                ;READ REVERSE ALL BLOCK EXCEPT BLOCK 1I01
004214  012777  004252  174342  TCRBK:  MOV     #TCRB1,@TCIV    ;SET UP INTERRUPT VECTOR
004222  012777  177400  174330          MOV     #-400,@TCWC     ;READ ONE BLOCK
```

```
004230  012777  004344  174324          MOV     #TCRBUF,@TCBA   ;WHERE BUFFER IS
004236  112777  000505  174306          MOVB    #IE+RD+DO,@TCCM ;READ THE BLOCK
004244  004767  000030                  JSR     %7,TC1          ;CHECK DATA BUFFER
004250  000002                          RTI                     ;EXIT - RETURN WHEN BLOCK IS READ
004252  005777  174274          TCRB11: TST     @TCCM           ;AND ERRORS
004256  100001                          BPL     .+4
004260  104006                          HLT                     ;DECTAPE ERROR
004262  012777  004140  174274          MOV     #TCR3,@TCIV     ;NEW VECTOR FOR BLOCK SEARCH
004270  112777  000502  174254          MOVB    #IE+RB,@TCCM    ;READ BLOCK FUNCTION
004276  000743                          BR      TCR4            ;RETURN TO BLOCK SEARCH

                                ;THIS ROUTINE CHECKS THE READ DATA BUFFER TC11
                                ;BY DOING A CHECK SUM ON THE DATA
004300  010146                  TC1:    MOV     %1,-(6)         ;SAVE THESE ON THE STACK
004302  010246                          MOV     %2,-(6)
004304  010346                          MOV     %3,-(6)
004306  005003                          CLR     %3              ;SUM OF DATA
004310  012701  004344                  MOV     #TCRBUF,%1      ;ADDRESS OF READ BUFFER
004314  012702  005344                  MOV     #TCRBUF+1000,%2 ;END OF READ BUFFER
004320  062103                  TC2:    ADD     (1)+,%3         ;EVEN ADD
004322  062103                          ADD     (1)+,%3         ;ODD ADD -2'S COMPLIMENT
004324  001401                          BEQ     .+4
004326  104006                          HLT                     ;DATA ERROR TC-11
004330  020102                          CMP     %1,%2           ;AT END OF BUFFER?
004332  001372                          BNE     TC2             ;NO - SUM THE REST
004334  012603                          MOV     (6)+,%3         ;RESTORE THE REGISTERS
004336  012602                          MOV     (6)+,%2
004340  012601                          MOV     (6)+,%1
004342  000207                          RTS     %7              ;EXIT

                                ;THIS WRITE BUFFER LOOK THE SAME FORWARD OR REVERSE
004344                          TCWBUF:
004344                          TCRBUF:
        000001                          N=1
004344  000001                          N                       ;DECTAPE WRITE BUFFER
004346  177777                          -N
        000002                          N=N+1
004350  000002                          N                       ;DECTAPE WRITE BUFFER
004352  177776                          -N
        000003                          N=N+1
004354  000003                          N                       ;DECTAPE WRITE BUFFER
004356  177775                          -N
        000004                          N=N+1
004360  000004                          N                       ;DECTAPE WRITE BUFFER
004362  177774                          -N
        000005                          N=N+1
004364  000005                          N                       ;DECTAPE WRITE BUFFER
004366  177773                          -N
        000006                          N=N+1
004370  000006                          N                       ;DECTAPE WRITE BUFFER
004372  177772                          -N
        000007                          N=N+1
004374  000007                          N                       ;DECTAPE WRITE BUFFER
004376  177771                          -N
```

```
                000010                  N=N+1
004400  000010                  N                       ;DECTAPE WRITE BUFFER
004402  177770                  -N
                000011                  N=N+1
004404  000011                  N                       ;DECTAPE WRITE BUFFER
004406  177767                  -N
                000012                  N=N+1
004410  000012                  N                       ;DECTAPE WRITE BUFFER
004412  177766                  -N
                000013                  N=N+1
004414  000013                  N                       ;DECTAPE WRITE BUFFER
004416  177765                  -N
                000014                  N=N+1
004420  000014                  N                       ;DECTAPE WRITE BUFFER
004422  177764                  -N
                000015                  N=N+1
004424  000015                  N                       ;DECTAPE WRITE BUFFER
004426  177763                  -N
                000016                  N=N+1
004430  000016                  N                       ;DECTAPE WRITE BUFFER
004432  177762                  -N
                000017                  N=N+1
004434  000017                  N                       ;DECTAPE WRITE BUFFER
004436  177761                  -N
                000020                  N=N+1
004440  000020                  N                       ;DECTAPE WRITE BUFFER
004442  177760                  -N
                000021                  N=N+1
004444  000021                  N                       ;DECTAPE WRITE BUFFER
004446  177757                  -N
                000022                  N=N+1
004450  000022                  N                       ;DECTAPE WRITE BUFFER
004452  177756                  -N
                000023                  N=N+1
004454  000023                  N                       ;DECTAPE WRITE BUFFER
004456  177755                  -N
                000024                  N=N+1
004460  000024                  N                       ;DECTAPE WRITE BUFFER
004462  177754                  -N
                000025                  N=N+1
004464  000025                  N                       ;DECTAPE WRITE BUFFER
004466  177753                  -N
                000026                  N=N+1
004470  000026                  N                       ;DECTAPE WRITE BUFFER
004472  177752                  -N
                000027                  N=N+1
004474  000027                  N                       ;DECTAPE WRITE BUFFER
004476  177751                  -N
                000030                  N=N+1
004500  000030                  N                       ;DECTAPE WRITE BUFFER
004502  177750                  -N
                000031                  N=N+1
004504  000031                  N                       ;DECTAPE WRITE BUFFER
004506  177747                  -N
```

```
                000032                  N=N+1
004510          000032                  N                       ;DECTAPE WRITE BUFFER
004512          177746                  -N
                000033                  N=N+1
004514          000033                  N                       ;DECTAPE WRITE BUFFER
004516          177745                  -N
                000034                  N=N+1
004520          000034                  N                       ;DECTAPE WRITE BUFFER
004522          177744                  -N
                000035                  N=N+1
004524          000035                  N                       ;DECTAPE WRITE BUFFER
004526          177743                  -N
                000036                  N=N+1
004530          000036                  N                       ;DECTAPE WRITE BUFFER
004532          177742                  -N
                000037                  N=N+1
004534          000037                  N                       ;DECTAPE WRITE BUFFER
004536          177741                  -N
                000040                  N=N+1
004540          000040                  N                       ;DECTAPE WRITE BUFFER
004542          177740                  -N
                000041                  N=N+1
004544          000041                  N                       ;DECTAPE WRITE BUFFER
004546          177737                  -N
                000042                  N=N+1
004550          000042                  N                       ;DECTAPE WRITE BUFFER
004552          177736                  -N
                000043                  N=N+1
004554          000043                  N                       ;DECTAPE WRITE BUFFER
004556          177735                  -N
                000044                  N=N+1
004560          000044                  N                       ;DECTAPE WRITE BUFFER
004562          177734                  -N
                000045                  N=N+1
004564          000045                  N                       ;DECTAPE WRITE BUFFER
004566          177733                  -N
                000046                  N=N+1
004570          000046                  N                       ;DECTAPE WRITE BUFFER
004572          177732                  -N
                000047                  N=N+1
004574          000047                  N                       ;DECTAPE WRITE BUFFER
004576          177731                  -N
                000050                  N=N+1
004600          000050                  N                       ;DECTAPE WRITE BUFFER
004602          177730                  -N
                000051                  N=N+1
004604          000051                  N                       ;DECTAPE WRITE BUFFER
004606          177727                  -N
                000052                  N=N+1
004610          000052                  N                       ;DECTAPE WRITE BUFFER
004612          177726                  -N
                000053                  N=N+1
004614          000053                  N                       ;DECTAPE WRITE BUFFER
004616          177725                  -N
```

```
                000054                  N=N+1
004620          000054                  N                       ;DECTAPE WRITE BUFFER
004622          177724                  -N
                000055                  N=N+1
004624          000055                  N                       ;DECTAPE WRITE BUFFER
004626          177723                  -N
                000056                  N=N+1
004630          000056                  N                       ;DECTAPE WRITE BUFFER
004632          177722                  -N
                000057                  N=N+1
004634          000057                  N                       ;DECTAPE WRITE BUFFER
004636          177721                  -N
                000060                  N=N+1
004640          000060                  N                       ;DECTAPE WRITE BUFFER
004642          177720                  -N
                000061                  N=N+1
004644          000061                  N                       ;DECTAPE WRITE BUFFER
004646          177717                  -N
                000062                  N=N+1
004650          000062                  N                       ;DECTAPE WRITE BUFFER
004652          177716                  -N
                000063                  N=N+1
004654          000063                  N                       ;DECTAPE WRITE BUFFER
004656          177715                  -N
                000064                  N=N+1
004660          000064                  N                       ;DECTAPE WRITE BUFFER
004662          177714                  -N
                000065                  N=N+1
004664          000065                  N                       ;DECTAPE WRITE BUFFER
004666          177713                  -N
                000066                  N=N+1
004670          000066                  N                       ;DECTAPE WRITE BUFFER
004672          177712                  -N
                000067                  N=N+1
004674          000067                  N                       ;DECTAPE WRITE BUFFER
004676          177711                  -N
                000070                  N=N+1
004700          000070                  N                       ;DECTAPE WRITE BUFFER
004702          177710                  -N
                000071                  N=N+1
004704          000071                  N                       ;DECTAPE WRITE BUFFER
004706          177707                  -N
                000072                  N=N+1
004710          000072                  N                       ;DECTAPE WRITE BUFFER
004712          177706                  -N
                000073                  N=N+1
004714          000073                  N                       ;DECTAPE WRITE BUFFER
004716          177705                  -N
                000074                  N=N+1
004720          000074                  N                       ;DECTAPE WRITE BUFFER
004722          177704                  -N
                000075                  N=N+1
004724          000075                  N                       ;DECTAPE WRITE BUFFER
004726          177703                  -N
```

```
                000076                  N=N+1
004730          000076                  N                       ;DECTAPE WRITE BUFFER
004732          177772                  -N
                000077                  N=N+1
004734          000077                  N                       ;DECTAPE WRITE BUFFER
004736          177771                  -N
                000100                  N=N+1
004740          000100                  N                       ;DECTAPE WRITE BUFFER
004742          177770                  -N
                000101                  N=N+1
                000100                  N=N-1
004744          177770                  -N
004746          000100                  N                       ;DEC TAPE WRITE BUFFER
                000077                  N=N-1
004750          177771                  -N
004752          000077                  N                       ;DEC TAPE WRITE BUFFER
                000076                  N=N-1
004754          177772                  -N
004756          000076                  N                       ;DEC TAPE WRITE BUFFER
                000075                  N=N-1
004760          177773                  -N
004762          000075                  N                       ;DEC TAPE WRITE BUFFER
                000074                  N=N-1
004764          177774                  -N
004766          000074                  N                       ;DEC TAPE WRITE BUFFER
                000073                  N=N-1
004770          177775                  -N
004772          000073                  N                       ;DEC TAPE WRITE BUFFER
                000072                  N=N-1
004774          177776                  -N
004776          000072                  N                       ;DEC TAPE WRITE BUFFER
                000071                  N=N-1
005000          177777                  -N
005002          000071                  N                       ;DEC TAPE WRITE BUFFER
                000070                  N=N-1
005004          177710                  -N
005006          000070                  N                       ;DEC TAPE WRITE BUFFER
                000067                  N=N-1
005010          177711                  -N
005012          000067                  N                       ;DEC TAPE WRITE BUFFER
                000066                  N=N-1
005014          177712                  -N
005016          000066                  N                       ;DEC TAPE WRITE BUFFER
                000065                  N=N-1
005020          177713                  -N
005022          000065                  N                       ;DEC TAPE WRITE BUFFER
                000064                  N=N-1
005024          177714                  -N
005026          000064                  N                       ;DEC TAPE WRITE BUFFER
                000063                  N=N-1
005030          177715                  -N
005032          000063                  N                       ;DEC TAPE WRITE BUFFER
                000062                  N=N-1
005034          177716                  -N
```

```
005036  000062                  N                       ;DEC TAPE WRITE BUFFER
        000061                  N=N-1
005040  177717                  -N
005042  000061                  N                       ;DEC TAPE WRITE BUFFER
        000060                  N=N-1
005044  177720                  -N
005046  000060                  N                       ;DEC TAPE WRITE BUFFER
        000057                  N=N-1
005050  177721                  -N
005052  000057                  N                       ;DEC TAPE WRITE BUFFER
        000056                  N=N-1
005054  177722                  -N
005056  000056                  N                       ;DEC TAPE WRITE BUFFER
        000055                  N=N-1
005060  177723                  -N
005062  000055                  N                       ;DEC TAPE WRITE BUFFER
        000054                  N=N-1
005064  177724                  -N
005066  000054                  N                       ;DEC TAPE WRITE BUFFER
        000053                  N=N-1
005070  177725                  -N
005072  000053                  N                       ;DEC TAPE WRITE BUFFER
        000052                  N=N-1
005074  177726                  -N
005076  000052                  N                       ;DEC TAPE WRITE BUFFER
        000051                  N=N-1
005100  177727                  -N
005102  000051                  N                       ;DEC TAPE WRITE BUFFER
        000050                  N=N-1
005104  177730                  -N
005106  000050                  N                       ;DEC TAPE WRITE BUFFER
        000047                  N=N-1
005110  177731                  -N
005112  000047                  N                       ;DEC TAPE WRITE BUFFER
        000046                  N=N-1
005114  177732                  -N
005116  000046                  N                       ;DEC TAPE WRITE BUFFER
        000045                  N=N-1
005120  177733                  -N
005122  000045                  N                       ;DEC TAPE WRITE BUFFER
        000044                  N=N-1
005124  177734                  -N
005126  000044                  N                       ;DEC TAPE WRITE BUFFER
        000043                  N=N-1
005130  177735                  -N
005132  000043                  N                       ;DEC TAPE WRITE BUFFER
        000042                  N=N-1
005134  177736                  -N
005136  000042                  N                       ;DEC TAPE WRITE BUFFER
        000041                  N=N-1
005140  177737                  -N
005142  000041                  N                       ;DEC TAPE WRITE BUFFER
        000040                  N=N-1
005144  177740                  -N
```

```
005146  000040                  N                       ;DEC TAPE WRITE BUFFER
        000037                  N=N-1
005150  177741                  -N
005152  000037                  N                       ;DEC TAPE WRITE BUFFER
        000036                  N=N-1
005154  177742                  -N
005156  000036                  N                       ;DEC TAPE WRITE BUFFER
        000035                  N=N-1
005160  177743                  -N
005162  000035                  N                       ;DEC TAPE WRITE BUFFER
        000034                  N=N-1
005164  177744                  -N
005166  000034                  N                       ;DEC TAPE WRITE BUFFER
        000033                  N=N-1
005170  177745                  -N
005172  000033                  N                       ;DEC TAPE WRITE BUFFER
        000032                  N=N-1
005174  177746                  -N
005176  000032                  N                       ;DEC TAPE WRITE BUFFER
        000031                  N=N-1
005200  177747                  -N
005202  000031                  N                       ;DEC TAPE WRITE BUFFER
        000030                  N=N-1
005204  177750                  -N
005206  000030                  N                       ;DEC TAPE WRITE BUFFER
        000027                  N=N-1
005210  177751                  -N
005212  000027                  N                       ;DEC TAPE WRITE BUFFER
        000026                  N=N-1
005214  177752                  -N
005216  000026                  N                       ;DEC TAPE WRITE BUFFER
        000025                  N=N-1
005220  177753                  -N
005222  000025                  N                       ;DEC TAPE WRITE BUFFER
        000024                  N=N-1
005224  177754                  -N
005226  000024                  N                       ;DEC TAPE WRITE BUFFER
        000023                  N=N-1
005230  177755                  -N
005232  000023                  N                       ;DEC TAPE WRITE BUFFER
        000022                  N=N-1
005234  177756                  -N
005236  000022                  N                       ;DEC TAPE WRITE BUFFER
        000021                  N=N-1
005240  177757                  -N
005242  000021                  N                       ;DEC TAPE WRITE BUFFER
        000020                  N=N-1
005244  177760                  -N
005246  000020                  N                       ;DEC TAPE WRITE BUFFER
        000017                  N=N-1
005250  177761                  -N
005252  000017                  N                       ;DEC TAPE WRITE BUFFER
        000016                  N=N-1
005254  177762                  -N
```

```
005256  000016                          N                       ;DEC TAPE WRITE BUFFER
        000015                          N-N-1
005260  177763                          -N
005262  000015                          N                       ;DEC TAPE WRITE BUFFER
        000014                          N-N-1
005264  177764                          -N
005266  000014                          N                       ;DEC TAPE WRITE BUFFER
        000013                          N-N-1
005270  177765                          -N
005272  000013                          N                       ;DEC TAPE WRITE BUFFER
        000012                          N-N-1
005274  177766                          -N
005276  000012                          N                       ;DEC TAPE WRITE BUFFER
        000011                          N-N-1
005300  177767                          -N
005302  000011                          N                       ;DEC TAPE WRITE BUFFER
        000010                          N-N-1
005304  177770                          -N
005306  000010                          N                       ;DEC TAPE WRITE BUFFER
        000007                          N-N-1
005310  177771                          -N
005312  000007                          N                       ;DEC TAPE WRITE BUFFER
        000006                          N-N-1
005314  177772                          -N
005316  000006                          N                       ;DEC TAPE WRITE BUFFER
        000005                          N-N-1
005320  177773                          -N
005322  000005                          N                       ;DEC TAPE WRITE BUFFER
        000004                          N-N-1
005324  177774                          -N
005326  000004                          N                       ;DEC TAPE WRITE BUFFER
        000003                          N-N-1
005330  177775                          -N
005332  000003                          N                       ;DEC TAPE WRITE BUFFER
        000002                          N-N-1
005334  177776                          -N
005336  000002                          N                       ;DEC TAPE WRITE BUFFER
        000001                          N-N-1
005340  177777                          -N
005342  000001                          N                       ;DEC TAPE WRITE BUFFER

005344  010701              BEGINX1:    MOV     PC,R1           ;SET UP R1 TO SELECT CURBNK
005346  042701  017777                  BIC     #17777,R1
005352  042737  160000  000034          BIC     #160000,@#34            ;SET SCOPE RET TO CURRENT BANK
005360  050137  000034                  BIS     R1,@#34
005364  000301                          SWAB    R1
005366  006201                          ASR     R1
005370  006201                          ASR     R1
005372  010137  000570                  MOV     R1,@#CURBNK

                                        ,SBTTL  BACKGROUND CPU TESTS
                                    ;BINARY INSTRUCTIONS
                                    ;INDEX, AND INDIRECT TEST OF PDP-11
005376  010767  007244          BEGIN1: MOV     PC,RETURN               ;FOR SCOPING - SETUP ADDRESS OF BEGIN1 IN
```

```
005402  062767  000014  007236          ADD     #14,RETURN              ;THIS BANK THRU CURRENT ASR
005410  012767  000400  007224          MOV     #400,ICOUNT             ;ITERATION COUNT
                                ;TEST COMPARE INSTRUCTION INDEXED
005416  012700  177770                  MOV     #-10,%0                 ;MINUS 10 TO REG 0
005422  026027  014660  125252          CMP     A(0),#125252            ;(A INDEX BY MINUS 10) TO #125252
005430  001401                          BEQ     .+4
005432  104006                          HLT                             ;COMPARE WITH INDEX FAILED
005434  104400                          SCOPE

005436  012700  000010                  MOV     #10,%0
005442  022760  052525  014660          CMP     #052525,A(0)
005450  001401                          BEQ     .+4
005452  104006                          HLT
005454  104400                          SCOPE

005456  012700  177770                  MOV     #-10,%0
005462  026060  014660  014660          CMP     A(0),A(0)
005470  001401                          BEQ     .+4
005472  104006                          HLT
005474  104400                          SCOPE

005476  012700  000010                  MOV     #+10,%0
005502  026060  014660  014660          CMP     A(0),A(0)
005510  001401                          BEQ     .+4
005512  104006                          HLT
005514  104400                          SCOPE

005516  012700  177774                  MOV     #-4,%0
005522  012701  000010                  MOV     #+10,%1
005526  026061  014660  014660          CMP     A(0),A(1)
005534  001401                          BEQ     .+4
005536  104006                          HLT
005540  104400                          SCOPE

005542  012700  177774                  MOV     #-4,%0
005546  012701  000010                  MOV     #10,%1
005552  026160  014660  014660          CMP     A(1),A(0)
005560  001401                          BEQ     .+4
005562  104006                          HLT
005564  104400                          SCOPE

                                ;TEST MOVE INSTRUCTION FOR INDEX

005566  012700  177770                  MOV     #-10,%0
005572  016067  014660  007102          MOV     A(0),TEMP
005600  026727  007076  125252          CMP     TEMP,#125252
005606  001401                          BEQ     .+4
005610  104006                          HLT
005612  104400                          SCOPE

005614  012700  177770                  MOV     #-10,%0
005620  012760  125252  014702          MOV     #125252,TEMP(0)
005626  023727  014672  125252          CMP     @#C,#125252
005634  001401                          BEQ     .+4
```

```
005636  104006                          HLT
005640  104400                          SCOPE

                                ;TEST BIC INSTRUCTION FOR INDEXING
005642  012767  177777  007032          MOV     #-1,TEMP
005650  012700  177770                  MOV     #-10,X0
005654  046067  014660  007020          BIC     A(0),TEMP
005662  026727  007014  052525          CMP     TEMP,#052525
005670  001401                          BEQ     .+4
005672  104006                          HLT
005674  104400                          SCOPE

005676  012700  177770                  MOV     #-10,X0
005702  012767  177777  006762          MOV     #-1,TEMP-10
005710  042767  052525  006754          BIC     #052525,TEMP-10
005716  026727  006750  125252          CMP     TEMP-10,#125252
005724  001401                          BEQ     .+4
005726  104006                          HLT
005730  104400                          SCOPE

005732  012737  125252  014702          MOV     #125252,@#TEMP
005740  012700  177770                  MOV     #-10,X0
005744  166760  006700  014712          SUB     B,TEMP+10(0)
005752  001401                          BEQ     .+4
005754  104006                          HLT
005756  104400                          SCOPE

005760  012737  052525  014702          MOV     #052525,@#TEMP
005766  012700  000010                  MOV     #10,X0
005772  166760  006672  014672          SUB     A+10,C(0)
006000  001401                          BEQ     .+4
006002  104006                          HLT
006004  104400                          SCOPE

                                ;TEST UNARYS INDEXED

006006  012737  177777  014702          MOV     #-1,@#TEMP
006014  012700  000010                  MOV     #+10,X0
006020  005060  014672                  CLR     C(0)
006024  005737  014702                  TST     @#TEMP
006030  001401                          BEQ     .+4
006032  104006                          HLT
006034  104400                          SCOPE

006036  012737  177777  014702          MOV     #-1,@#TEMP
006044  012700  000010                  MOV     #10,X0
006050  005160  014672                  COM     C(0)
006054  005737  014702                  TST     @#TEMP
006060  001401                          BEQ     .+4
006062  104006                          HLT
006064  104400                          SCOPE

006066  012737  177777  014702          MOV     #-1,@#TEMP
006074  012700  177770                  MOV     #-10,X0
```

```
006100  005260  014712                  INC     D(0)
006104  005737  014702                  TST     @#TEMP
006110  001401                          BEQ     .+4
006112  104006                          HLT
006114  104400                          SCOPE

006116  012737  000001  014702          MOV     #1,@#TEMP
006124  012700  177770                  MOV     #-10,%0
006130  005360  014712                  DEC     D(0)
006134  005737  014702                  TST     @#TEMP
006140  001401                          BEQ     .+4
006142  104006                          HLT
006144  104400                          SCOPE

006146  012737  000001  014702          MOV     #1,@#TEMP
006154  012700  000010                  MOV     #10,%0
006160  005360  014672                  DEC     C(0)
006164  005737  014702                  TST     @#TEMP
006170  001401                          BEQ     .+4
006172  104006                          HLT
006174  104400                          SCOPE

006176  012737  000001  014702          MOV     #1,@#TEMP
006204  012700  177770                  MOV     #-10,%0
006210  005460  014712                  NEG     D(0)
006214  022737  177777  014702          CMP     #-1,@#TEMP
006222  001401                          BEQ     .+4
006224  104006                          HLT
006226  104400                          SCOPE

006230  012737  000001  014702          MOV     #1,@#TEMP
006236  012700  000010                  MOV     #+10,%0
006242  005460  014672                  NEG     C(0)
006246  022737  177777  014702          CMP     #-1,@#TEMP
006254  001401                          BEQ     .+4
006256  104006                          HLT
006260  104400                          SCOPE

006262  012737  177777  014702          MOV     #-1,@#TEMP
006270  012700  177770                  MOV     #-10,%0
006274  000261                          SEC
006276  005560  014712                  ADC     D(0)
006302  005737  014702                  TST     @#TEMP
006306  001401                          BEQ     .+4
006310  104006                          HLT
006312  104400                          SCOPE

006314  012737  177777  014702          MOV     #-1,@#TEMP
006322  012700  000010                  MOV     #+10,%0
006326  000261                          SEC
006330  005560  014672                  ADC     C(0)
006334  005737  014702                  TST     @#TEMP
006340  001401                          BEQ     .+4
006342  104006                          HLT
```

```
006344  104400                          SCOPE

006346  012737  000001  014702          MOV     #1,@#TEMP
006354  012700  177770                  MOV     #-10,%0
006360  000261                          SEC
006362  005660  014712                  SBC     D(0)
006366  005737  014702                  TST     @#TEMP
006372  001401                          BEQ     .+4
006374  104006                          HLT
006376  104400                          SCOPE

006400  012737  000001  014702          MOV     #1,@#TEMP
006406  012700  000010                  MOV     #+10,%0
006412  000261                          SEC
006414  005660  014672                  SBC     C(0)
006420  005737  014702                  TST     @#TEMP
006424  001401                          BEQ     .+4
006426  104006                          HLT
006430  104400                          SCOPE

                                ;TEST JMP INDIRECT
006432  010700                          MOV     %7,%0
006434  062700  000010                  ADD     #10,%0
006440  000110                          JMP     @%0
006442  104006                          HLT
006444  000240                          NOP
006446  104400                          SCOPE

006450  010700                          MOV     %7,%0
006452  062700  000010                  ADD     #10,%0
006456  000110                          JMP     @%0
006460  104006                          HLT
006462  000240                          NOP
006464  104400                          SCOPE

                                ;TEST INDIRECT ADDRESSING
                                ;TEST COMPARE INSTRUCTION
006466  023727  014650  125252          CMP     @#B,#125252
006474  001401                          BEQ     .+4
006476  104006                          HLT
006500  104400                          SCOPE

006502  022737  125252  014650          CMP     #125252,@#B
006510  001401                          BEQ     .+4
006512  104006                          HLT
006514  104400                          SCOPE

006516  023737  014650  014650          CMP     @#B,@#B
006524  001401                          BEQ     .+4
006526  104006                          HLT
006530  104400                          SCOPE

                                ;TEST MOVE INSTRUCTIONS
006532  013700  014650                  MOV     @#B,%0
```

```
006536  022700  125252                  CMP     #125252,X0
006542  001401                          BEQ     .+4
006544  104006                          HLT
006546  104400                          SCOPE

006550  012737  125252  014702          MOV     #125252,@#TEMP
006556  023737  014650  014702          CMP     @#B,@#TEMP
006564  001401                          BEQ     .+4
006566  104006                          HLT
006570  104400                          SCOPE

006572  013737  014650  014672          MOV     @#B,@#C
006600  023737  014650  014672          CMP     @#B,@#C
006606  001401                          BEQ     .+4
006610  104006                          HLT
006612  104400                          SCOPE

                                ;TEST BIC INSTRUCTION INDIRECT
006614  012700  177777                  MOV     #-1,X0
006620  043700  014650                  BIC     @#B,X0
006624  020027  052525                  CMP     X0,#052525
006630  001401                          BEQ     .+4
006632  104006                          HLT
006634  104400                          SCOPE

006636  012737  177777  014702          MOV     #-1,@#TEMP
006644  042737  125252  014702          BIC     #125252,@#TEMP
006652  022737  052525  014702          CMP     #052525,@#TEMP
006660  001401                          BEQ     .+4
006662  104006                          HLT
006664  104400                          SCOPE

006666  012737  177777  014672          MOV     #-1,@#C
006674  043737  014650  014672          BIC     @#B,@#C
006702  023727  014672  052525          CMP     @#C,#52525
006710  001401                          BEQ     .+4
006712  104006                          HLT
006714  104400                          SCOPE

                                ;TEST SUBTRACT INSTRUCTION
006716  012700  125252                  MOV     #125252,X0
006722  163700  014650                  SUB     @#B,X0
006726  020027  000000                  CMP     X0,#0
006732  001401                          BEQ     .+4
006734  104006                          HLT
006736  104400                          SCOPE

006740  012737  125252  014702          MOV     #125252,@#TEMP
006746  166737  005676  014702          SUB     B,@#TEMP
006754  001401                          BEQ     .+4
006756  104006                          HLT
006760  104400                          SCOPE

006762  012767  125252  005712          MOV     #125252,TEMP
```

```
007770  163767  014650  005704          SUB     @#B,TEMP
007776  005767  005700                  TST     TEMP
010002  001401                          BEQ     .+4
010004  104006                          HLT
010006  104400                          SCOPE

                                ;TEST ADD INDIRECT
010010  005000                          CLR     X0
010012  063700  014650                  ADD     @#B,X0
010016  022700  125252                  CMP     #125252,X0
010022  001401                          BEQ     .+4
010024  104006                          HLT
010026  104400                          SCOPE

010030  005037  014702                  CLR     @#TEMP
010034  062737  125252  014702          ADD     #125252,@#TEMP
010042  022737  125252  014702          CMP     #125252,@#TEMP
010050  001401                          BEQ     .+4
010052  104006                          HLT
010054  104400                          SCOPE

010056  012737  125252  014702          MOV     #125252,@#TEMP
010064  067737  005576  014702          ADD     @A+6,@#TEMP
010072  023727  014702  177777          CMP     @#TEMP,#-1
010100  001401                          BEQ     .+4
010102  104006                          HLT
010104  104400                          SCOPE

                                ;TEST UNARYS INDIRECT
010106  012737  177777  014702          MOV     #-1,@#TEMP
010114  005037  014702                  CLR     @#TEMP
010120  005737  014702                  TST     @#TEMP
010124  001401                          BEQ     .+4
010126  104006                          HLT
010130  104400                          SCOPE

010132  012737  125252  014702          MOV     #125252,@#TEMP
010140  005137  014702                  COM     @#TEMP
010144  022737  052525  014702          CMP     #052525,@#TEMP
010152  001401                          BEQ     .+4
010154  104006                          HLT
010156  104400                          SCOPE

010160  005037  014702                  CLR     @#TEMP
010164  005237  014702                  INC     @#TEMP
010170  022737  000001  014702          CMP     #1,@#TEMP
010176  001401                          BEQ     .+4
010200  104006                          HLT
010202  104400                          SCOPE

010204  005037  014702                  CLR     @#TEMP
010210  005377  005470                  DEC     @TEMP+2
010214  023727  014702  177777          CMP     @#TEMP,#-1
010222  001401                          BEQ     .+4
```

```
007224  104006                          HLT
007226  104400                          SCOPE

007230  012737  000001  014702          MOV     #1,@#TEMP
007236  005437  014702                  NEG     @#TEMP
007242  022737  177777  014702          CMP     #-1,@#TEMP
007250  001401                          BEQ     .+4
007252  104006                          HLT
007254  104400                          SCOPE

                                ;TEST INDIRECT ADDRESSING WITH INDEXING
                                ;TEST COMPARE INSTRUCTION
007256  027727  005370  125252          CMP     @B+2,#125252
007264  001401                          BEQ     .+4
007266  104006                          HLT
007270  104400                          SCOPE

007272  022777  125252  005352          CMP     #125252,@B+2
007300  001401                          BEQ     .+4
007302  104006                          HLT
007304  104400                          SCOPE

007306  027777  005340  005336          CMP     @B+2,@B+2
007314  001401                          BEQ     .+4
007316  104006                          HLT
007320  104400                          SCOPE

                                ;TEST MOVE INSTRUCTIONS
007322  017700  005324                  MOV     @B+2,R0
007326  022700  125252                  CMP     #125252,R0
007332  001401                          BEQ     .+4
007334  104006                          HLT
007336  104400                          SCOPE

007340  012777  125252  005336          MOV     #125252,@TEMP+2
007346  023737  014650  014702          CMP     @#B,@#TEMP
007354  001401                          BEQ     .+4
007356  104006                          HLT
007360  104400                          SCOPE

007362  017777  005264  005304          MOV     @B+2,@C+2
007370  023737  014650  014672          CMP     @#B,@#C
007376  001401                          BEQ     .+4
007400  104006                          HLT
007402  104400                          SCOPE

                                ;TEST BIC INSTRUCTION INDIRECT WITH INDEXING
007404  012700  177777                  MOV     #-1,R0
007410  047700  005236                  BIC     @B+2,R0
007414  020027  052525                  CMP     R0,#52525
007420  001401                          BEQ     .+4
007422  104006                          HLT
007424  104400                          SCOPE
```

```
007426  012737  177777  014702          MOV     #-1,@#TEMP
007434  042777  125252  005242          BIC     #125252,@TEMP+2
007442  022737  052525  014702          CMP     #52525,@#TEMP
007450  001401                          BEQ     .+4
007452  104006                          HLT
007454  104400                          SCOPE

007456  012737  177777  014672          MOV     #-1,@#C
007464  047777  005162  005202          BIC     @B+2,@C+2
007472  026737  005172  014672          CMP     A+10,@#C
007500  001401                          BEQ     .+4
007502  104006                          HLT
007504  104400                          SCOPE

007506  012700  125252                  MOV     #125252,X0
007512  167700  005134                  SUB     @B+2,X0
007516  020027  000000                  CMP     X0,#0
007522  001401                          BEQ     .+4
007524  104006                          HLT
007526  104400                          SCOPE

007530  012737  125252  014702          MOV     #125252,@#TEMP
007536  166777  005106  005140          SUB     B,@TEMP+2
007544  001401                          BEQ     .+4
007546  104006                          HLT
007550  104400                          SCOPE

007552  012737  125252  014702          MOV     #125252,@#TEMP
007560  167777  005066  005116          SUB     @B+2,@TEMP+2
007566  005737  014702                  TST     @#TEMP
007572  001401                          BEQ     .+4
007574  104006                          HLT
007576  104400                          SCOPE

                                ;TEST ADD INDIRECT WITH INDEXING
007600  005000                          CLR     X0
007602  067700  005044                  ADD     @B+2,X0
007606  022700  125252                  CMP     #125252,X0
007612  001401                          BEQ     .+4
007614  104006                          HLT
007616  104400                          SCOPE

007620  005037  014702                  CLR     @#TEMP
007624  062777  125252  005052          ADD     #125252,@TEMP+2
007632  022737  125252  014702          CMP     #125252,@#TEMP
007640  001401                          BEQ     .+4
007642  104006                          HLT
007644  104400                          SCOPE

007646  012737  125252  014702          MOV     #125252,@#TEMP
007654  067777  005006  005022          ADD     @A+6,@TEMP+2
007662  023727  014702  177777          CMP     @#TEMP,#-1
007670  001401                          BEQ     .+4
007672  104006                          HLT
```

```
007674  104400                          SCOPE

                                ;TEST UNARYS INDIRECT WITH INDEXING
007676  012737  177777  014702          MOV     #-1,@#TEMP
007704  005077  004774                  CLR     @TEMP+2
007710  005737  014702                  TST     @#TEMP
007714  001401                          BEQ     .+4
007716  104006                          HLT
007720  104400                          SCOPE

007722  012737  125252  014702          MOV     #125252,@#TEMP
007730  005177  004750                  COM     @TEMP+2
007734  022737  052525  014702          CMP     #052525,@#TEMP
007742  001401                          BEQ     .+4
007744  104006                          HLT
007746  104400                          SCOPE

007750  005037  014702                  CLR     @#TEMP
007754  005277  004724                  INC     @TEMP+2
007760  022737  000001  014702          CMP     #1,@#TEMP
007766  001401                          BEQ     .+4
007770  104006                          HLT
007772  104400                          SCOPE

007774  005037  014702                  CLR     @#TEMP
010000  005377  004700                  DEC     @TEMP+2
010004  023727  014702  177777          CMP     @#TEMP,#-1
010012  001401                          BEQ     .+4
010014  104006                          HLT
010016  104400                          SCOPE

010020  012737  000001  014702          MOV     #1,@#TEMP
010026  005477  004652                  NEG     @TEMP+2
010032  022737  177777  014702          CMP     #-1,@#TEMP
010040  001401                          BEQ     .+4
010042  104006                          HLT
010044  104400                          SCOPE

010046  012737  177777  014702          MOV     #-1,@#TEMP
010054  000261                          SEC
010056  005577  004622                  ADC     @TEMP+2
010062  005737  014702                  TST     @#TEMP
010066  001401                          BEQ     .+4
010070  104006                          HLT
010072  104400                          SCOPE

010074  012737  000001  014702          MOV     #1,@#TEMP
010102  000261                          SEC
010104  005677  004574                  SBC     @TEMP+2
010110  005737  014702                  TST     @#TEMP
010114  001401                          BEQ     .+4
010116  104006                          HLT
010120  104400                          SCOPE
```

```
                                        ;TEST OF COMBINED INDEXING AND INDIRECT
010122  012700  177772                          MOV     #-6,X0
010126  027027  014660  125252                  CMP     @A(0),#125252
010134  001401                                  BEQ     .+4
010136  104006                                  HLT
010140  104400                                  SCOPE

010142  012700  177772                          MOV     #-6,X0
010146  022770  125252  014660                  CMP     #125252,@A(0)
010154  001401                                  BEQ     .+4
010156  104006                                  HLT
010160  104400                                  SCOPE

010162  012700  177772                          MOV     #-6,X0
010166  012701  000002                          MOV     #+2,X1
010172  027071  014660  014660                  CMP     @A(0),@A(1)
010200  001401                                  BEQ     .+4
010202  104006                                  HLT
010204  104400                                  SCOPE

                                        ;TEST BIC INSTRUCTION
010206  012700  000006                          MOV     #+6,X0
010212  012767  177777  004462                  MOV     #-1,TEMP
010220  047067  014660  004454                  BIC     @A(0),TEMP
010226  022767  125252  004446                  CMP     #125252,TEMP
010234  001401                                  BEQ     .+4
010236  104006                                  HLT
010240  104400                                  SCOPE

010242  012700  177772                          MOV     #-6,X0
010246  012737  177777  014672                  MOV     #-1,@#C
010254  042770  125252  014702                  BIC     #125252,@TEMP(0)
010262  023727  014672  052525                  CMP     @#C,#052525
010270  001401                                  BEQ     .+4
010272  104006                                  HLT
010274  104400                                  SCOPE

010276  012737  177777  014672                  MOV     #-1,@#C
010304  012700  177772                          MOV     #-6,X0
010310  012701  177772                          MOV     #-6,X1
010314  047071  014660  014702                  BIC     @A(0),@TEMP(1)
010322  022737  052525  014672                  CMP     #052525,@#C
010330  001401                                  BEQ     .+4
010332  104006                                  HLT
010334  104400                                  SCOPE

                                        ;BINARY INSTRUCTIONS
                                        ;INDEX, AND INDIRECT TEST OF PDP-11
                                        ;TEST COMPARE INSTRUCTION INDEXED
010336  012700  177770                          MOV     #-10,X0                 ;MINUS 10 TO REG 0
010342  126027  014660  000252                  CMPB    A(0),#000252            ;(A INDEX BY MINUS 10) TO #125252
010350  001401                                  BEQ     .+4
010352  104006                                  HLT                             ;COMPARE WITH INDEX FAILED
010354  104400                                  SCOPE
```

```
010356  012700  177770          MOV     #-10,%0                 ;FOR INDEX
010362  122760  000252  014660  CMPB    #000252,A(0)            ;A INDEXED
010370  001401                  BEQ     .+4
010372  104006                  HLT
010374  104400                  SCOPE

010376  012700  000010          MOV     #10,%0                  ;INDEX
010402  126027  014660  000125  CMPB    A(0),#000125
010410  001401                  BEQ     .+4
010412  104006                  HLT
010414  104400                  SCOPE

010416  012700  000010          MOV     #10,%0
010422  122760  000125  014660  CMPB    #000125,A(0)
010430  001401                  BEQ     .+4
010432  104006                  HLT
010434  104400                  SCOPE

010436  012700  177770          MOV     #-10,%0
010442  126060  014660  014660  CMPB    A(0),A(0)
010450  001401                  BEQ     .+4
010452  104006                  HLT
010454  104400                  SCOPE

010456  012700  000010          MOV     #+10,%0
010462  126060  014660  014660  CMPB    A(0),A(0)
010470  001401                  BEQ     .+4
010472  104006                  HLT
010474  104400                  SCOPE

010476  012700  177770          MOV     #-10,%0
010502  012701  000004          MOV     #+4,%1
010506  126061  014660  014660  CMPB    A(0),A(1)
010514  001401                  BEQ     .+4
010516  104006                  HLT
010520  104400                  SCOPE

010522  126160  014660  014660  CMPB    A(1),A(0)
010530  001401                  BEQ     .+4
010532  104006                  HLT
010534  104400                  SCOPE

010536  012700  177774          MOV     #-4,%0
010542  012701  000010          MOV     #+10,%1
010546  126061  014660  014660  CMPB    A(0),A(1)
010554  001401                  BEQ     .+4
010556  104006                  HLT
010560  104400                  SCOPE

010562  012700  177774          MOV     #-4,%0
010566  012701  000010          MOV     #10,%1
010572  126160  014660  014660  CMPB    A(1),A(0)
010600  001401                  BEQ     .+4
```

```
010602  104006                          HLT
010604  104400                          SCOPE

                                ;TEST MOVE INSTRUCTION FOR INDEX
010606  012700  177778                  MOV     #-10,R0
010612  116067  014660  004062          MOVB    A(R0),TEMP
010620  126727  004056  000252          CMPB    TEMP,#000252
010626  001401                          BEQ     .+4
010630  104006                          HLT
010632  104400                          SCOPE

010634  012700  000010                  MOV     #+10,R0
010640  116067  014660  004034          MOVB    A(R0),TEMP
010646  126727  004030  000125          CMPB    TEMP,#000125
010654  001401                          BEQ     .+4
010656  104006                          HLT
010660  104400                          SCOPE

010662  012700  177778                  MOV     #-10,R0
010666  112760  125252  014702          MOVB    #125252,TEMP(R0)
010674  123727  014672  125252          CMPB    @#C,#125252
010702  001401                          BEQ     .+4
010704  104006                          HLT
010706  104400                          SCOPE

010710  012700  000010                  MOV     #+10,R0
010714  112760  052525  014702          MOVB    #052525,TEMP(R0)
010722  123727  014712  052525          CMPB    @#TEMP+10,#052525
010730  001401                          BEQ     .+4
010732  104006                          HLT
010734  104400                          SCOPE

                                ;TEST BIC INSTRUCTION FOR INDEXING
010736  012767  177777  003736          MOV     #-1,TEMP
010744  012700  177778                  MOV     #-10,R0
010750  146067  014660  003724          BICB    A(R0),TEMP
010756  126727  003720  177525          CMPB    TEMP,#177525
010764  001401                          BEQ     .+4
010766  104006                          HLT
010770  104400                          SCOPE

010772  012767  177777  003702          MOV     #-1,TEMP
011000  012700  000010                  MOV     #10,R0
011004  146067  014660  003670          BICB    A(R0),TEMP
011012  126727  003664  007652          CMPB    TEMP,#007652
011020  001401                          BEQ     .+4
011022  104006                          HLT
011024  104400                          SCOPE

011026  012737  177777  014712          MOV     #-1,@#TEMP+10
011034  012700  000010                  MOV     #10,R0
011040  142760  125252  014702          BICB    #125252,TEMP(R0)
011046  123727  014712  002525          CMPB    @#TEMP+10,#2525
011054  001401                          BEQ     .+4
```

```
011056  104006                          HLT
011060  104400                          SCOPE

011062  012700  177770                  MOV     #-10,%0
011066  012767  177777  003576          MOV     #-1,TEMP-10
011074  142767  052525  003570          BICB    #052525,TEMP-10
011102  126727  003564  125252          CMPB    TEMP-10,#125252
011110  001401                          BEQ     .+4
011112  104006                          HLT
011114  104400                          SCOPE

                                ;TEST UNARYS INDEXED
011116  012737  177777  014702          MOV     #-1,@#TEMP
011124  012700  177770                  MOV     #-10,%0
011130  105060  014712                  CLRB    D(0)
011134  105737  014702                  TSTB    @#TEMP
011140  001401                          BEQ     .+4
011142  104006                          HLT
011144  104400                          SCOPE

011146  012737  177777  014702          MOV     #-1,@#TEMP
011154  012700  177770                  MOV     #-10,%0
011160  105060  014712                  CLRB    D(0)
011164  023727  014702  177400          CMP     @#TEMP,#177400
011172  001401                          BEQ     .+4
011174  104006                          HLT
011176  104400                          SCOPE

011200  012737  177777  014702          MOV     #-1,@#TEMP
011206  012700  177771                  MOV     #-7,%0
011212  105060  014712                  CLRB    D(0)
011216  023727  014702  000377          CMP     @#TEMP,#000377
011224  001401                          BEQ     .+4
011226  104006                          HLT
011230  104400                          SCOPE

011232  012737  177777  014702          MOV     #-1,@#TEMP
011240  012700  000010                  MOV     #+10,%0
011244  105060  014672                  CLRB    C(0)
011250  105737  014702                  TSTB    @#TEMP
011254  001401                          BEQ     .+4
011256  104006                          HLT
011260  104400                          SCOPE

011262  012737  177777  014702          MOV     #-1,@#TEMP
011270  012700  177770                  MOV     #-10,%0
011274  105160  014712                  COMB    D(0)
011300  105737  014702                  TSTB    @#TEMP
011304  001401                          BEQ     .+4
011306  104006                          HLT
011310  104400                          SCOPE

011312  012737  177777  014702          MOV     #-1,@#TEMP
011320  012700  000010                  MOV     #+10,%0
```

```
011324  105260  014672                  INCB    C(0)
011330  105737  014702                  TSTB    @#TEMP
011334  001401                          BEQ     .+4
011336  104006                          HLT
011340  104400                          SCOPE

011342  012737  000001  014702          MOV     #1,@#TEMP
011350  012700  177770                  MOV     #-10,%0
011354  105360  014712                  DECB    D(0)
011360  105737  014702                  TSTB    @#TEMP
011364  001401                          BEQ     .+4
011366  104006                          HLT
011370  104400                          SCOPE

011372  012737  000001  014702          MOV     #1,@#TEMP
011400  012700  000010                  MOV     #+10,%0
011404  105460  014672                  NEGB    C(0)
011410  023727  014702  000377          CMP     @#TEMP,#377
011416  001401                          BEQ     .+4
011420  104006                          HLT
011422  104400                          SCOPE

011424  012737  177777  014702          MOV     #-1,@#TEMP
011432  012700  177770                  MOV     #-10,%0
011436  000261                          SEC
011440  105560  014712                  ADCB    D(0)
011444  023727  014702  177400          CMP     @#TEMP,#177400
011452  001401                          BEQ     .+4
011454  104006                          HLT
011456  104400                          SCOPE

011460  012737  000001  014702          MOV     #1,@#TEMP
011466  012700  000010                  MOV     #+10,%0
011472  000261                          SEC
011474  105660  014672                  SBCB    C(0)
011500  005737  014702                  TST     @#TEMP
011504  001401                          BEQ     .+4
011506  104006                          HLT
011510  104400                          SCOPE

                                ;TEST INDIRECT ADDRESSING
                                ;TEST COMPARE INSTRUCTION
011512  123727  014650  000252          CMPB    @#B,#000252
011520  001401                          BEQ     .+4
011522  104006                          HLT
011524  104400                          SCOPE

011526  122737  125252  014650          CMPB    #125252,@#B
011534  001401                          BEQ     .+4
011536  104006                          HLT
011540  104400                          SCOPE

                                ;TEST MOVE INSTRUCTIONS
011542  113700  014650                  MOVB    @#B,%0
```

```
011546  122700  000252                  CMPB    #000252,X0
011552  001401                          BEQ     .+4
011554  104006                          HLT
011556  104400                          SCOPE

011560  112737  125252  014702          MOVB    #125252,@#TEMP
011566  126737  003056  014702          CMPB    B,@#TEMP
011574  001401                          BEQ     .+4
011576  104006                          HLT
011600  104400                          SCOPE

                                ;TEST UNARYS INDIRECT
011602  012737  177777  014702          MOV     #-1,@#TEMP
011610  105037  014702                  CLRB    @#TEMP
011614  023727  014702  177400          CMP     @#TEMP,#177400
011622  001401                          BEQ     .+4
011624  104006                          HLT
011626  104400                          SCOPE

011630  012737  125252  014702          MOV     #125252,@#TEMP
011636  105137  014703                  COMB    @#TEMP+1
011642  022737  052652  014702          CMP     #052652,@#TEMP
011650  001401                          BEQ     .+4
011652  104006                          HLT
011654  104400                          SCOPE

011656  005037  014702                  CLR     @#TEMP
011662  105237  014703                  INCB    @#TEMP+1
011666  022737  000400  014702          CMP     #400,@#TEMP
011674  001401                          BEQ     .+4
011676  104006                          HLT
011700  104400                          SCOPE

011702  005037  014702                  CLR     @#TEMP
011706  105377  002772                  DECB    @TEMP+2
011712  023727  014702  000377          CMP     @#TEMP,#377
011720  001401                          BEQ     .+4
011722  104006                          HLT
011724  104400                          SCOPE

011726  005037  014702                  CLR     @#TEMP
011732  112737  000001  014703          MOVB    #1,@#TEMP+1
011740  105437  014703                  NEGB    @#TEMP+1
011744  022737  177400  014702          CMP     #177400,@#TEMP
011752  001401                          BEQ     .+4
011754  104006                          HLT
011756  104400                          SCOPE

                                ;TEST INDIRECT ADDRESSING WITH INDEXING
                                ;TEST COMPARE INSTRUCTION

011760  122777  125252  002664          CMPB    #125252,@B+2
011766  001401                          BEQ     .+4
011770  104006                          HLT
```

```
012772  104400                          SCOPE

012774  127777  002652  002650          CMPB    @B+2,@B+2
013002  001401                          BEQ     .+4
013004  104006                          HLT
013006  104400                          SCOPE

                                ;TEST MOVE INSTRUCTIONS
013010  117700  002636                  MOVB    @B+2,X0
013014  122700  125252                  CMPB    #125252,X0
013020  001401                          BEQ     .+4
013022  104006                          HLT
013024  104400                          SCOPE

013026  112777  125252  002650          MOVB    #125252,@TEMP+2
013034  126737  002610  014702          CMPB    B,@#TEMP
013042  001401                          BEQ     .+4
013044  104006                          HLT
013046  104400                          SCOPE

013050  117777  002576  002616          MOVB    @B+2,@C+2
013056  126737  002566  014672          CMPB    B,@#C
013064  001401                          BEQ     .+4
013066  104006                          HLT
013070  104400                          SCOPE

                                ;TEST BIC INSTRUCTION INDIRECT WITH INDEXING
013072  012700  177777                  MOV     #-1,X0
013076  147700  002550                  BICB    @B+2,X0
013102  120027  052525                  CMPB    X0,#52525
013106  001401                          BEQ     .+4
013110  104006                          HLT
013112  104400                          SCOPE

013114  012737  177777  014702          MOV     #-1,@#TEMP
013122  142777  125252  002554          BICB    #125252,@TEMP+2
013130  122737  052525  014702          CMPB    #52525,@#TEMP
013136  001401                          BEQ     .+4
013140  104006                          HLT
013142  104400                          SCOPE

013144  012737  177777  014672          MOV     #-1,@#C
013152  147777  002474  002514          BICB    @B+2,@C+2
013160  126737  002504  014672          CMPB    A+10,@#C
013166  001401                          BEQ     .+4
013170  104006                          HLT
013172  104400                          SCOPE

                                ;TEST UNARYS INDIRECT WITH INDEXING
013174  012737  177777  014702          MOV     #-1,@#TEMP
013202  105077  002476                  CLRB    @TEMP+2
013206  105737  014702                  TSTB    @#TEMP
013212  001401                          BEQ     .+4
013214  104006                          HLT
```

```
012216  104400                          SCOPE

012220  005037  014702                  CLR     @#TEMP
012224  105277  002454                  INCB    @TEMP+2
012230  122737  000001  014702          CMPB    #1,@#TEMP
012236  001401                          BEQ     .+4
012240  104006                          HLT
012242  104400                          SCOPE

012244  005037  014702                  CLR     @#TEMP
012250  105377  002430                  DECB    @TEMP+2
012254  123727  014702  177777          CMPB    @#TEMP,#-1
012262  001401                          BEQ     .+4
012264  104006                          HLT
012266  104400                          SCOPE

012270  012737  000001  014702          MOV     #1,@#TEMP
012276  105477  002402                  NEGB    @TEMP+2
012302  122737  177777  014702          CMPB    #-1,@#TEMP
012310  001401                          BEQ     .+4
012312  104006                          HLT
012314  104400                          SCOPE

012316  012737  177777  014702          MOV     #-1,@#TEMP
012324  000261                          SEC
012326  105577  002352                  ADCB    @TEMP+2
012332  022737  177400  014702          CMP     #177400,@#TEMP
012340  001401                          BEQ     .+4
012342  104006                          HLT
012344  105737  014702                  TSTB    @#TEMP
012350  001401                          BEQ     .+4
012352  104006                          HLT
012354  104400                          SCOPE

012356  012737  000001  014702          MOV     #1,@#TEMP
012364  000261                          SEC
012366  105377  002312                  DECB    @TEMP+2
012372  005737  014702                  TST     @#TEMP
012376  001401                          BEQ     .+4
012400  104006                          HLT
012402  104400                          SCOPE

                                ;TEST OF COMBINED INDEXING AND INDIRECT
012404  012700  177772                  MOV     #-6,%0
012410  127027  014660  125252          CMPB    @A(0),#125252
012416  001401                          BEQ     .+4
012420  104006                          HLT
012422  104400                          SCOPE

012424  012700  177772                  MOV     #-6,%0
012430  012701  000002                  MOV     #+2,%1
012434  127071  014660  014660          CMPB    @A(0),@A(1)
012442  001401                          BEQ     .+4
012444  104006                          HLT
```

```
012446  104400                          SCOPE

                                ;TEST BIC INSTRUCTION
012450  012700  000006                  MOV     #+6,X0
012454  012767  177777  002220          MOV     #-1,TEMP
012462  147067  014660  002212          BICB    @A(0),TEMP
012470  122767  125252  002204          CMPB    #125252,TEMP
012476  001401                          BEQ     .+4
012500  104006                          HLT
012502  104400                          SCOPE

012504  012700  177772                  MOV     #-6,X0
012510  012737  177777  014672          MOV     #-1,@#C
012516  142770  125252  014702          BICB    #125252,@TEMP(0)
012524  123727  014672  000125          CMPB    @#C,#000125
012532  001401                          BEQ     .+4
012534  104006                          HLT
012536  104400                          SCOPE

012540  012700  014652                  MOV     #B+2,X0         ;ADDRESS OF ADDRESS OF B
012544  023067  002100                  CMP     @(0)+,B
012550  001401                          BEQ     .+4
012552  104006                          HLT
012554  104400                          SCOPE

012556  012700  014654                  MOV     #B+4,X0
012562  023067  002062                  CMP     @-(0),B
012566  001401                          BEQ     .+4
012570  104006                          HLT
012572  104400                          SCOPE

012574  012700  014654                  MOV     #B+4,X0
012600  125067  002044                  CMPB    @-(0),B
012604  001401                          BEQ     .+4
012606  104006                          HLT
012610  104400                          SCOPE

012612  012700  014676                  MOV     #C+4,X0
012616  012737  177777  014672          MOV     #-1,@#C
012624  105050                          CLRB    @-(0)
012626  023727  014672  177400          CMP     @#C,#177400
012634  001401                          BEQ     .+4
012636  104006                          HLT
012640  104400                          SCOPE

012642  012737  177777  014672          MOV     #-1,@#C
012650  012700  177772                  MOV     #-6,X0
012654  012701  177772                  MOV     #-6,X1
012660  147071  014660  014702          BICB    @A(0),@TEMP(1)
012666  022737  177525  014672          CMP     #177525,@#C
012674  001401                          BEQ     .+4
012676  104006                          HLT
012700  104400                          SCOPE
```

```
                                        ;SET UP TO TEST THAT R0 IS NOT DESTROYED BY FALSE SELECTION
012702  012700  052525                          MOV     #52525,%0       ;THIS IS CHECKED LATER IN PROGRAM

                                        ;TEST JSR INSTRUCTION
012706  004767  000002                          JSR     %7, TJSR2       ;PLACE PC ON STACK
012712  000405                        TJSR1:  BR      TJSR3           ;RETURN HERE ON RTS %19
012714  121627  012712                TJSR2:  CMPB    @%6,#TJSR1      ;CHECK FOR CORRECT PC ON STACK
012720  001401                                BEQ     .+4
012722  104006                                  HLT                     ;INCORRECT PC ON STACK
012724  000207                                  RTS     %7              ;RETURN TO IMST AFTER JSR
012726  104400                        TJSR3:  SCOPE

012730  000257                                  CCC
012732  004717                                  JSR     %7,@%7          ;INSTRUCTION UNDER TEST
012734  121627  012734                          CMPB    @%6,#TJSR3+6            ;TEST THE STACK
012740  001401                                BEQ     .+4
012742  104006                                  HLT                     ;PC OF JSR DID NOT GO TO STACK
012744  005726                                  TST     (6)+            ;REPOSITION THE STACK
012746  104400                                  SCOPE

                                        ;TEST NESTED SUBROUTINES
012750  000257                                  CCC                     ;CLEAR CONDITION CODES
012752  004767  001564                          JSR     %7,SUBR6
012756  100401                                  BMI     .+4
012760  104006                                  HLT                     ;JSR OR RTS FAILED
012762  001401                                  BEQ     .+4
012764  104006                                  HLT                     ;JSR OR RTS FAILED
012766  102401                                  BVS     .+4
012770  104006                                  HLT                     ;JSR OR RTS FAILED
012772  103401                                  BCS     .+4
012774  104006                                  HLT                     ;JSR OR RTS FAILED
012776  104400                                  SCOPE

                                        ;TEST ROTATE ODD BYTE
013000  104400                                  SCOPE
013002  000257                                  CCC                             ;CLEAR "C"
013004  012767  123456  001670                  MOV     #123456,TEMP
013012  106067  001665                          RORB    TEMP+1                  ;ROTATE ODD BYTE
013016  103401                                  BCS     .+4
013020  104006                                  HLT                             ;C NOT SET
013022  102401                                  BVS     .+4
013024  104006                                  HLT                             ;V NOT SET
013026  022767  051456  001646                  CMP     #051456,TEMP
013034  001401                                  BEQ     .+4
013036  104006                                  HLT                             ;ROTATE FAILED
013040  104400                                  SCOPE

013042  000277                                  SCC                             ;SET C
013044  012767  123456  001630                  MOV     #123456,TEMP
013052  106067  001625                          RORB    TEMP+1
013056  103401                                  BCS     .+4
013060  104006                                  HLT                             ;C NOT SET
013062  102001                                  BVC     .+4
013064  104006                                  HLT                             ;V NOT CLEARED
```

```
013066  022767  151456  001606          CMP     #151456,TEMP
013074  001401                          BEQ     .+4
013076  104006                          HLT
013100  104400                          SCOPE                   ;ROTATE FAILED

013102  000257                          CCC
013104  012767  123456  001570          MOV     #123456,TEMP
013112  106167  001565                  ROLB    TEMP+1
013116  103401                          BCS     .+4
013120  104006                          HLT                     ;C NOT SET
013122  102401                          BVS     .+4
013124  104006                          HLT                     ;V NOT SET
013126  022767  047056  001546          CMP     #047056,TEMP
013134  001401                          BEQ     .+4
013136  104006                          HLT
013140  104400                          SCOPE                   ;ROTATE BYTE FAILED

013142  000277                          SCC                     ;SET C
013144  012767  123456  001530          MOV     #123456,TEMP
013152  106167  001525                  ROLB    TEMP+1
013156  103401                          BCS     .+4
013160  104006                          HLT
013162  102401                          BVS     .+4             ;C NOT SET
013164  104006                          HLT                     ;V NOT SET
013166  022767  047456  001506          CMP     #047456,TEMP
013174  001401                          BEQ     .+4
013176  104006                          HLT
013200  104400                          SCOPE                   ;ROTATE ODD BYTE FAILED

013202  000257                          CCC                     ;CLEAR C
013204  012767  177777  001470          MOV     #-1,TEMP
013212  106267  001465                  ASRB    TEMP+1
013216  103401                          BCS     .+4
013220  104006                          HLT
013222  102001                          BVC     .+4             ;C NOT SET
013224  104006                          HLT
013226  026727  001450  177777          CMP     TEMP,#-1        ;V NOT CLEARED
013234  001401                          BEQ     .+4
013236  104006                          HLT
013240  104400                          SCOPE                   ;SHIFT FAILED

013242  000277                          SCC
013244  012767  177777  001430          MOV     #-1,TEMP
013252  106367  001425                  ASLB    TEMP+1
013256  103401                          BCS     .+4
013260  104006                          HLT                     ;C NOT SET
013262  102001                          BVC     .+4
013264  104006                          HLT                     ;V NOT CLEARED
013266  026727  001410  177377          CMP     TEMP,#177377
013274  001401                          BEQ     .+4
013276  104006                          HLT
013300  104400                          SCOPE                   ;SHIFT BYTE FAILED
```

```
                                        ;TEST THAT R0 WASN'T CLEARED BY FALSE SELECTION
013302  022700  052525                          CMP     #52525,R0
013306  001401                                  BEQ     .+4
013310  104006                                  HLT
013312  104400                                  SCOPE

                                        ;TEST COMBINATION OF N, C AND V
013314  005067  001322                          CLR     ICOUNT                  ;NO ITERATION

                                        ;;INHIBIT TESTS WHICH USE ALL NUMBERS WHEN SW11 IS SET
013320  032737  004000  000550                  BIT     #4000,@#SREG2
013326  001402                                  BEQ     COMPAR
013330  000167  001152                          JMP     DONE

                                        ;TEST ALL COMBINATIONS OF NUMBERS WITH COMPARE INSTRUCTION
013334  005000                          COMPAR: CLR     %0                      ;;INIT %0
013336  005001                                  CLR     %1                      ;INIT %1
013340  020001                          CMP1:   CMP     %0,%1                   ;ARE THE EQUAL
013342  001401                                  BEQ     .+4
013344  104006                                  HLT                             ;R0 AND R1 DID NOT COMPARE
013346  020027  177777                          CMP     %0,#-1                  ;AT UPPER LIMIT
013352  001403                                  BEQ     CMP2                    ;YES EXIT
013354  005200                                  INC     %0                      ;;INCREMENT TO NEXT NUMBER
013356  005201                                  INC     %1
013360  000767                                  BR      CMP1
013362  104400                          CMP2:   SCOPE

                                        ;TEST ROTATING ALL NUMBERS
013364  104400                                  SCOPE
013366  012767  177777  000132                  MOV     #-1,REFF                ;;INITIALIZE BASE NUMBER
013374  005267  000126                  TSROT:  INC     REFF                    ;;INCREMENT NUMBER
013400  004767  000012                          JSR     %7,ROTALL               ;GO TO COMPARE ROUTINE
013404  026727  000116  177777                  CMP     REFF,#-1                ;TEST ALL VALUES
013412  001370                                  BNE     TSROT                   ;NO TEST THEM ALL
013414  000446                                  BR      TSRT2A                  ;WE ARE DONE

013416  016767  000104  000104  ROTALL: MOV     REFF,TEST
013424  006067  000100                  ROR     TEST
013430  006067  000074                  ROR     TEST
013434  006067  000070                  ROR     TEST
013440  006167  000064                  ROL     TEST
013444  006167  000060                  ROL     TEST
013450  006167  000054                  ROL     TEST
013454  100004                          BPL     .+12
013456  103007                          BCC     .+20                    ;Z=1
013460  102013                          BVC     .+30                    ;Z=1, C=1
013462  104006                          HLT                             ;Z=C, BUT V=1
013464  000411                          BR      .+24
013466  103006                          BCC     .+16                    ;Z=0
013470  102407                          BVS     .+20                    ;Z=0, C=1
013472  104006                          HLT                             ;Z NOT EQUAL C, V=1
013474  000405                          BR      .+14
013476  102404                          BVS     .+12                    ;Z=1, C=0
013500  104006                          HLT                             ;Z NOT EQUAL C, V=1
```

```
013502  000402                          BR      .+6
013504  102001                          BVC     .+4                     ;Z=0, C=0
013506  104006                          HLT                             ;Z=C, BUT V=1
013510  104400                          SCOPE
013512  026767  000012  000006          CMP     TEST,REFF
013520  001401                          BEQ     .+4
013522  104006                          HLT                             ;INITIAL NOT EQUAL TO FINAL
013524  000207                          RTS     %7
013526  000000          REFF:   0
013530  000000          TEST:   0
        013526                  REFF=REFF

                                ;TEST ROTATING BYTE EVEN/ODD, ALL NUMBERS
013532  012767  177777  177766  TSRT2A: MOV     #-1,REFF
013540  005267  177762          TSROT2: INC     REFF
013544  004767  000016                  JSR     %7,ROTBE
013550  004767  000122                  JSR     %7,ROTBO
013554  022767  177777  177744          CMP     #-1,REFF
013562  001366                          BNE     TSROT2
013564  000505                          BR      ROTEN1
013566  016767  177734  177734  ROTBE:  MOV     REFF,TEST
013574  106067  177730                  RORB    TEST                    ;ROTATE BYTE EVEN
013600  106067  177724                  RORB    TEST
013604  106067  177720                  RORB    TEST
013610  106167  177714                  ROLB    TEST
013614  106167  177710                  ROLB    TEST
013620  106167  177704                  ROLB    TEST
013624  100004                          BPL     .+12
013626  103007                          BCC     .+20                    ;Z=1
013630  102013                          BVC     .+30                    ;Z=1, C=1
013632  104006                          HLT                             ;Z=C, BUT V=1
013634  000411                          BR      .+24
013636  103006                          BCC     .+16                    ;Z=0
013640  102407                          BVS     .+20                    ;Z=0, C=1
013642  104006                          HLT                             ;Z NOT EQUAL C, V=1
013644  000405                          BR      .+14
013646  102404                          BVS     .+12                    ;Z=1, C=0
013650  104006                          HLT                             ;Z NOT EQUAL C, V=1
013652  000402                          BR      .+6
013654  102001                          BVC     .+4                     ;Z=0, C=0
013656  104006                          HLT                             ;Z=C, BUT V=1
013660  104400                          SCOPE
013662  026767  177642  177636          CMP     TEST,REFF
013670  001401                          BEQ     .+4
013672  104006                          HLT
013674  000207                          RTS     %7
013676  106067  177627          ROTBO:  RORB    TEST+1                  ;ROTATE BYTE ODD
013702  106067  177623                  RORB    TEST+1
013706  106067  177617                  RORB    TEST+1
013712  106167  177613                  ROLB    TEST+1
013716  106167  177607                  ROLB    TEST+1
013722  106167  177603                  ROLB    TEST+1
013726  100004                          BPL     .+12
013730  103007                          BCC     .+20                    ;Z=1
```

```
013732  102013                          BVC     ,+30                    ;Z=1, C=1
013734  104006                          HLT                             ;Z=C, BUT V=1
013736  000411                          BR      ,+24
013740  103006                          BCC     ,+16                    ;Z=0
013742  102407                          BVS     ,+20                    ;Z=0, C=1
013744  104006                          HLT                             ;Z NOT EQUAL C, V=1
013746  000405                          BR      ,+14
013750  102404                          BVS     ,+12                    ;Z=1, C=0
013752  104006                          HLT                             ;Z NOT EQUAL C, V=1
013754  000402                          BR      ,+6
013756  102001                          BVC     ,+4                     ;Z=0, C=0
013760  104006                          HLT                             ;Z=C, BUT V=1
013762  104400                          SCOPE
013764  026767  177540  177534         CMP     TEST,REFF
013772  001401                          BEQ     ,+4
013774  104006                          HLT
013776  000207                          RTS     %7
014000  104400              RO%EN1:     SCOPE

                                    ;ADD AND SUBTRACT ALL NUMBERS AGAINST FIXED NUMBERS
                                    ;A+B=C, C=A+B, BF SHOULD EQUAL B!
014002  011667  000072                  MOV     @%6,NUMA
014006  012767  000001  177512         MOV     #1,REF
014014  005267  177506      ARITST:     INC     REF
014020  004767  000014                  JSR     %7,AD$UB
014024  022767  177777  177474         CMP     #-1,REFF
014032  001370                          BNE     ARITST
014034  000422                          BR      ARIEND
014036  104400                          SCOPE
014040  016767  177462  177462  ADSUB:  MOV     REF,TEST
014046  066767  000026  177454         ADD     NUMA,TEST
014054  166767  000020  177446         SUB     NUMA,TEST
014062  026767  177440  177440         CMP     REF,TEST
014070  001401                          BEQ     ,+4
014072  104006                          HLT
014074  104400                          SCOPE
014076  000207                          RTS     %7
014100  000000              NUMA:       0
014102  104400              ARIEND:     SCOPE

                                    ;TEST COMPLIMENTING ALL NUMBERS
014104  005067  000572                  CLR     TEMP                    ;BASE DATA
014110  005067  000572                  CLR     TEMP+4                  ;BASE REFERENCE
014114  005167  000562      TCOM:       COM     TEMP                    ;COMPLIMENT DATA
014120  005367  000562                  DEC     TEMP+4                  ;DECREMENT REFERENCE
014124  026767  000552  000554         CMP     TEMP,TEMP+4             ;COMPARE
014132  001401                          BEQ     ,+4                     ;TEST
014134  104006                          HLT                             ;COMPLIMENT OR DECREMENT FAILED
014136  005167  000540                  COM     TEMP
014142  005267  000534                  INC     TEMP                    ;INCREMENT AND TEST FOR DONE
014146  001362                          BNE     TCOM                    ;NOT FINISHED GO LOOP
014150  104400                          SCOPE

                                    ;TEST COMB (EVEN BYTE)
```

```
014152  005067  000524                 CLR     TEMP            ;BASE DATA
014156  005067  000524                 CLR     TEMP+4          ;REFERENCE DATA
014162  105167  000514         TCOM2:  COMB    TEMP
014166  005367  000514                 DEC     TEMP+4
014172  126767  000504  000506         CMPB    TEMP,TEMP+4     ;COMPARE
014200  001401                         BEQ     .+4
014202  104006                         HLT                     ;COMPLIMENT OR INCREMENT BYTE FAILED
014204  105167  000472                 COMB    TEMP
014210  105267  000466                 INCB    TEMP
014214  001362                         BNE     TCOM2
014216  104400                         SCOPE

                                ;TEST COMB (ODD BYTE)
014220  005067  000456                 CLR     TEMP            ;BASE DATA
014224  005067  000456                 CLR     TEMP+4          ;REFERENCE DATA
014230  105167  000447         TCOM3:  COMB    TEMP+1          ;ODD BYTE
014234  005367  000446                 DEC     TEMP+4
014240  126767  000437  000440         CMPB    TEMP+1,TEMP+4
014246  001401                         BEQ     .+4
014250  104006                         HLT                     ;COMPLIMENT BYTE FAILED
014252  105167  000425                 COMB    TEMP+1
014256  105267  000421                 INCB    TEMP+1
014262  001362                         BNE     TCOM3
014264  104400                         SCOPE

                                ;TEST COMPARE ALL VALUE EVEN BYTE WITH ODD
014266  005067  000410                 CLR     TEMP            ;BASE VALUE
014272  126767  000404  000403 TSCOMB: CMPB    TEMP,TEMP+1     ;COMPARE
014300  001401                         BEQ     .+4
014302  104006                         HLT                     ;COMPARE FAILED
014304  002001                         BGE     .+4
014306  104006                         HLT                     ;V IS NOT = TO N
014310  003401                         BLE     .+4
014312  104006                         HLT                     ;V IS SET
014314  062767  000401  000360         ADD     #401,TEMP
014322  022767  177777  000352         CMP     #-1,TEMP
014330  001360                         BNE     TSCOMB
014332  104400                         SCOPE

014334  012767  000010  000300         MOV     #10,ICOUNT

                                ;TEST TO SEE IF I/O DEVICES WERE SELECTED
014342  123727  000550  000377         CMPB    @#SREG2,#377    ;SELECTED DEVICES STORED IN SREG2
014350  001404                         BEQ     WAIT4           ;BRANCH IF NO DEVICES SELECTED
014352  000001                         WAIT                    ;INTERRUPTS WILL OCCUR
014354  000001                         WAIT                    ;IF DEVICES ARE SELECTED
014356  000001                         WAIT
014360  000001                         WAIT
014362  104400                 WAIT4:  SCOPE
014364  012767  000400  000250         MOV     #400,ICOUNT

                                ;TEST SWAB
014372  012767  000200  177130         MOV     #0200,TEST
014400  000367  177124                 SWAB    TEST
```

```
014404  100001                          BPL     .+4
014406  104006                          HLT
014410  001401                          BEQ     .+4
014412  104006                          HLT
014414  000367  177110                  SWAB    TEST
014420  100401                          BMI     .+4
014422  104006                          HLT
014424  001001                          BNE     .+4
014426  104006                          HLT
014430  104400                          SCOPE

                                ;TEST ALL COMBINATIONS OF SWAB
014432  005067  177072                  CLR     TEST            ;NUMBER UNDER TEST
014436  005067  177064                  CLR     REF             ;REFERENCE NUMBER
014442  000367  177062          SWABA1: SWAB    TEST            ;OPERATION UNDER TEST
014446  026767  177056  177052          CMP     TEST,REF        ;TEST SWAB INSTRUCTION
014454  001401                          BEQ     .+4
014456  104006                          HLT                     ;SWAB FAILED
014460  000367  177044                  SWAB    TEST
014464  005267  177036                  INC     REF             ;INCREMENT REFERENCE NUMBER
014470  105267  177035                  INCB    TEST+1          ;INC TEST NUMBER
014474  001362                          BNE     SWABA           ;LOOP TILL DONE
014476  104400                          SCOPE
014500  012767  004000  000134          MOV     #4000,ICOUNT


                                ;*****************************************************************
                                ;END OF USER CODE IN BANK/
                                ;CALL KERNEL/
                                ;ALTERRED IN CORE EXPANSION/
014506  104016                  DONE1:  EOB
014510  000240                          NOP                     ;TO ALLOW CORE EXPANSION TO PATCH IN JMP

                                ;GROUP OF NESTED SUBROUTINES/
014512  000207                  SUBR1:  RTS     %7              ;ONE INSTRUCTION
014514  000277                  SUBR2:  SCC                     ;ONE DEEP
014516  000207                          RTS     %7
014520  004767  177770          SUBR3:  JSR     %7,SUBR2        ;TWO DEEP
014524  000207                          RTS     %7
014526  004767  177766          SUBR4:  JSR     %7,SUBR3        ;THREE DEEP
014532  000207                          RTS     %7
014534  004767  177766          SUBR5:  JSR     %7,SUBR4        ;FOUR DEEP
014540  000207                          RTS     %7
014542  004767  177766          SUBR6:  JSR     %7,SUBR5        ;FIVE DEEP
014546  000207                          RTS     %7

                                ;SCOPE AND/OR ITERATION LOOP FOR EACH TEST TIMES/
014550  032767  002000  163772  SCOPEC: BIT     #2000,SREG2     ;INHIBIT PROCESSOR TESTS?
014556  001403                          BEQ     .+10            ;NO
014560  022626                          CMP     (SP)+,(SP)+
014562  000167  165454                  JMP     MAIN            ;YES
014566  032767  040000  163754          BIT     #40000,SREG2    ;TEST SR FOR SCOPE
014574  001012                          BNE     SCOPE8          ;YES, SCOPE
014576  032767  004000  163744          BIT     #4000,SREG2     ;NO-TEST FOR ITERATION
```

```
014604  001011                          BNE     SCOPEG          ;INHIBIT ITERATION
014606  026767  000032  000026          CMP     SCOPEF,ICOUNT   ;COMPARE CURRENT COUNT TO MAX NUMBER
014614  100005                          BPL     SCOPEG          ;EXIT-DONE
014616  005267  000022                  INC     SCOPEF          ;INCREMENT COUNT
014622  016716  000020          SCOPEB: MOV     RETURN,@SP
014626  000002                          RTI
014630  005067  000010          SCOPEG: CLR     SCOPEF          ;CLEAR COUNT
014634  011667  000006                  MOV     @X6,RETURN      ;SAVE SCOPE RETURN POINTER
014640  000002                          RTI                     ;RETURN INLINE-NEXT TEST
014642  000400                  ICOUNT: 400                     ;ITERATION COUNT
014644  000000                  SCOPEF: 0                       ;COUNT LOCATION FOR ITERATION LOOP
014646  000000                  RETURN: 0                       ;ADDRESS OF LAST TEST

                                ;FIXED VALUES FOR USE IN TEST/
014650  125252                  B:      125252
014652  014650                          B                       ;ADDRESS OF B
014654  052525                          052525
        014660                          .-B+10
014660  177777                  A:      -1
014662  014664                          A+4
        014664                          .-A+4
014664  125252                          125252
014666  014670                          A+10                    ;ADDRESS OF A+10
014670  052525                          052525

                                ;FOR STORAGE
014672  000000                  C:      0
014674  014672                          C                       ;ADDRESS OF C
        014702                          .-C+10
014702  000000                  TEMP:   0
014704  014702                          TEMP                    ;ADDRESS OF TEMP
        014710                          .-TEMP+6
014710  014712                          TEMP+10                 ;ADDRESS OF TEMP+10 OR "D"
014712  000000                  D:      0

                                ;SUBROUTINE TO INITIALIZE ALL PAGES TO NR, BANK 0, 1 PAGE, UP/
014714  010146                  NRALL:  MOV     R1,-(R0)        ;SAVE REGISTERS
014716  010246                          MOV     R2,-(R0)
014720  010346                          MOV     R3,-(R0)
014722  012701  000536                  MOV     #IPDRTAB,R1     ;R1 HOLDS ADDRESS OF CURRENT POSITION
                                                                ;IN TABLE OF ADDRESSES
014726  012703  000010          NRLOOP: MOV     #8.,R3          ;R3 USED AS COUNTER
014732  012102                          MOV     (R1)+,R2        ;R2 CONTAINS ADDRESS OF PDR OR
                                                                ;PAR TO BE CLEARED
014734  005022                          CLR     (R2)+           ;CLEAR ALL ASR'S FOR THIS MODE
014736  077302                          SOB     R3,.-2
014740  020127  000544                  CMP     R1,#IPDREND     ;CHECK FOR DONE
014744  003770                          BLE     NRLOOP          ;CLEAR ALL IN NEXT MODE IF NOT DONE
014746  012603                          MOV     (R6)+,R3
014750  012602                          MOV     (R6)+,R2
014752  012601                          MOV     (R6)+,R1
014754  000207                          RTS     X7

                                ;EMT HANDLER/
```

```
                                        ;FIRST 3 CALLS LEFT OPEN IN TABLE FOR EASY PATCHES/
015756  162716  000002          EM$SRV: SUB     #2,@SP                  ;GET CALL
015762  006576  000000                  MFPI    @(SP)
015766  012667  000022                  MOV     (SP)+,EPC
015772  062716  000002                  ADD     #2,@SP
015776  105067  000013                  CLRB    EPC+1                   ;SAVE OFFSET ONLY
016002  062767  015016  000004          ADD     #EMTAB,EPC              ;POINT TO TABLE OF ADDRESSES
016010  017767  000000                  MOV     @EPC,PC                 ;JUMP TO DESIRED ROUTINE
016014  000000                  EPC:    0
        000000                          PATCH1=0
        000000                          PATCH2=0
        000000                          PATCH3=0
016016  000000          EM$TAB: PATCH1                                  ;PATCH IN ADDRESS OF ROUTINE
016020  000000                  PATCH2
016022  000000                  PATCH3
016024  016072                  PRINT                                   ;ERROR PRINTOUT
016026  016038                  EOBSRV                                  ;END OF BANK

                                        ;END OF BANK SERVICE
016030  113737  177571  000551  EOBSRV: MOVB    @#SR+1,@#SREG2+1        ;READ SWITCHES AGAIN
016036  032767  000001  163502          BIT     #1,SREG1        ;KT11-D INHIBITED?
016044  001507                  BEQ     EOB2            ;NO = CONTINUE
016046  004767  001410          EOB1:   JSR     %7,BELL         ;SIGNAL END OF PASS
016052  042766  000020  000002          BIC     #20,2(SP)       ;CLEAR TRACE BIT OF STATUS ON STACK
016060  012737  000016  000014          MOV     #16,@#14        ;SETUP TRACE RETURN TO CAUSE HALT
016066  005037  000016                  CLR     @#16            ;IF A TRACE TRAP OCCURS
016072  032737  010000  000550          BIT     #10000,@#SREG2  ;INHIBIT TRACE TRAPPING?
016100  001011                  BNE     EOB1A           ;YES = BRANCH
016102  005167  163474                  COM     TRPB            ;SWITCH TRACE FLAG
016106  100006                  BPL     EOB1A           ;IF NOT SET, LEAVE TRACE OFF
016110  052766  000020  000002          BIS     #20,2(SP)       ;IF SET, SET TRACE BIT OF STATUS ON STACK
016116  012737  016070  000014          MOV     #TRTRP,@#14
016124  032737  000040  000546  EOB1A: BIT     #40,@#SREG1     ;CORE EXPANSION INHIBITED?
016132  001051                  BNE     EOB1C           ;YES, SKIP
016134  013701  000550                  MOV     @#SREG2,R1
016140  032767  000002  163436          BIT     #2,MEM0
016146  001402                  BEQ     DSW1
016150  010137  020550                  MOV     R1,@#SREG2+20000
016154  032767  000004  163422  DSW1:   BIT     #4,MEM0
016162  001402                  BEQ     DSW2
016164  010137  040550                  MOV     R1,@#SREG2+40000
016170  032767  000010  163406  DSW2:   BIT     #10,MEM0
016176  001402                  BEQ     DSW3
016200  010137  060550                  MOV     R1,@#SREG2+60000
016204  032767  000020  163372  DSW3:   BIT     #20,MEM0
016212  001402                  BEQ     DSW4
016214  010137  100550                  MOV     R1,@#SREG2+100000
016220  032767  000040  163356  DSW4:   BIT     #40,MEM0
016226  001402                  BEQ     DSW5
016230  010137  120550                  MOV     R1,@#SREG2+120000
016234  032767  000100  163342  DSW5:   BIT     #100,MEM0
016242  001402                  BEQ     EOB1B
016244  010137  140550                  MOV     R1,@#SREG2+140000
016250  012716  005344          EOB1B:  MOV     #BEGINX,(SP)
```

```
015254  000002                          RTI
015256  012716  005376          EOB1C:  MOV     #BEGIN,(SP)
015262  000002                          RTI
015264  042737  000340  177776  EOB2:   BIC     #340,@#PSR
015272  032767  000002  163246          BIT     #2,SREG1        ;USER/KERNEL INHIBITED?
015300  001262                          BNE     EOB1            ;YES - SET PC AND RETURN
015302  032767  000004  163236          BIT     #4,SREG1        ;NO--INHIBIT 4K AS 32K?
015310  001256                          BNE     EOB1            ;YES - SET PC AND RETURN
015312  026767  163256  163174          CMP     CURPAR,UPAR7    ;LAST USER ASR DONE?
015320  001444                          BEQ     NXTBNK          ;YES - GO FIND NEXT BANK
015322  062737  020000  000034          ADD     #20000,@#34     ;UPDATE SCOPE VECTOR ADDRESS IN BANK 0
015330  062767  020000  163242          ADD     #20000,BNKSTR   ;UPDATE BANK START TO REFERENCE CURRENT ASR
015336  016716  163236                  MOV     BNKSTR,(SP)
015342  026767  163142  163224          CMP     UPAR0,CURPAR
015350  001404                          BEQ     NXTSEG
015352  005077  163216                  CLR     @CURPAR         ;SET PREVIOUS ASR TO NR, BANK 0
015356  005077  163214                  CLR     @CURPDR
015362  062767  000002  163204  NXTSEG: ADD     #2,CURPAR       ;UPDATE POINTERS TO NEXT SEGMENT
015370  062767  000002  163200          ADD     #2,CURPDR
015376  012777  077406  163172          MOV     #77406,@CURPDR  ;SET NEXT SEGMENT RW, 4K
015404  016777  163160  163162          MOV     CURBNK,@CURPAR  ;MAP NEXT SEGMENT TO CURRENT BANK
015412  052737  030000  177776          BIS     #30000,@#PSR    ;SET PREVIOUS MODE TO USER
015420  006506                          MFPI    R6              ;PICK UP USER STACK POINTER
015422  062716  020000                  ADD     #20000,@R6      ;MAP IT TO NEXT ASR
015426  006606                          MTPI    R6              ;PUT IT BACK
015430  000002                          RTI                     ;GO BACK TO MAINLINE
015432  005327  000000          NXTBNK: DEC     #0              ;STALL SO DOUBLE BELL NOTED
015436  001375                          BNE     .-4
015440  004767  001016                  JSR     %7,BELL
015444  012746  000400                  MOV     #UBUFF,-(SP)
015450  052737  030000  177776          BIS     #30000,@#PSR
015456  006606                          MTPI    R6
015460  013737  000570  000572          MOV     @#CURBNK,@#OLDBNK       ;SAVE PREV BANK ADDRESS
015466  062767  000200  163074  BNKTST: ADD     #200,CURBNK
015474  006367  163110                  ASL     COREPT
015500  103006                          BCC     1$
015502  012767  000001  163100          MOV     #1,COREPT
015510  012767  000606  163074          MOV     #MEM1,MEMUT
015516  022767  007600  163044  1$:     CMP     #7600,CURBNK    ;CHECK FOR EXTERNAL BANK
015524  001067                          BNE     EOB3            ;IF NOT, TEST FOR ITS PRESENCE
015526  012767  000000  163034          MOV     #0,CURBNK       ;START OVER, TESTING BANK 0
015534  012767  000001  163046          MOV     #1,COREPT
015542  012767  000604  163042          MOV     #MEM0,MEMUT
015550  013701  000042          LOGIC:  MOV     @#42,R1
015554  001412                          BEQ     BNKT
015556  000005                          RESET
015560  005046                          CLR     -(SP)           ;CLEAR TBIT VIA RTI
015562  012746  015570                  MOV     #LOGICAL,-(SP)
015566  000002                          RTI
015570  004711          LOGICAL:JSR     %7,@R1
015572  000240                          NOP
015574  000240                          NOP
015576  000240                          NOP
015600  000000                          HALT
```

```
015602  032737  000001  000550  BNKT1:  BIT     #1,@#SREG2      ;TTY OUT SELECTED
015610  001410                          BEQ     BNKT1           ;YES, NO ASTERISK
015612  004767  000662                  JSR     %7,CRLF
015616  105777  162566                  TSTB    @TCSR           ;WAIT FOR TELETYPE
015622  100375                          BPL     .-4
015624  012777  000252  162560          MOV     #252,@TDBR      ;OUTPUT ASTERISK TO SIGNAL END OF PASS
015632  042766  000020  000006  BNKT11: BIC     #20,6(SP)       ;CLEAR TRACE BIT OF STATUS ON STACK
015640  012737  000016  000014          MOV     #16,@#14
015646  005037  000016                  CLR     @#16
015652  032737  010000  177570          BIT     #10000,@#SR
015660  001011                          BNE     EOB3
015662  005167  162714                  COM     TRPB
015666  100006                          BPL     EOB3
015670  052766  000020  000006          BIS     #20,6(SP)
015676  012737  016070  000014          MOV     #TRTRP,@#14
015704  016777  162660  162620  EOB3:   MOV     CURBNK,@KPAR2   ;MAP KERNEL SEGMENT 2 TO BANK BEING LOOKED FOR
015712  012777  077406  162602          MOV     #77406,@KPDR2
015720  036777  162664  162664          BIT     COREPT,@MEMUT
015726  001657                          BEQ     BNKTST
015730  042737  160000  000034          BIC     #160000,@#34    ;INITIALIZE SCOPE VECTOR ADDRESS
015736  005001                          CLR     R1              ;R1 ADDRESSES BANK 0 THRU KERNEL ASR0
015740  012702  040000                  MOV     #40000,R2       ;R2 ADDRESSES NEW BANK THRU KERNEL ASR2
015744  012703  014712                  MOV     #D,R3
015750  006203                          ASR     R3
015752  012122                  CORMOV: MOV     (R1)+,(R2)+
015754  077302                          SOB     R3,CORMOV
015756  016767  162526  162610          MOV     UPAR0,CURPAR    ;FIRST ASR CHECKED IS USER ASR0
015764  016767  162512  162604          MOV     UPDR0,CURPDR
015772  016777  162572  162574          MOV     CURBNK,@CURPAR
016000  012777  077406  162570          MOV     #77406,@CURPDR
016006  005077  162502                  CLR     @UPAR7
016012  005077  162478                  CLR     @UPDR7
016016  026727  162550  000000          CMP     OLDBNK,#0               ;PREV BANK = 0
016024  001414                          BEQ     EOB6            ;YES, DO NOT CLEAR
016026  016777  162540  162476          MOV     OLDBNK,@KPAR2
016034  012777  077406  162460          MOV     #77406,@KPDR2
016042  012701  040000                  MOV     #40000,R1
016046  012703  007630                  MOV     #7630,R3
016052  005021                  BNKLP:  CLR     (R1)+
016054  077302                          SOB     R3,BNKLP
016056  012716  005376          EOB6:   MOV     #BEGIN,(SP)
016062  011667  162512                  MOV     (SP),BNKSTR
016066  000002                          RTI

                                ;RTT EXECUTED WHEN TRACE IS ON/
016070  000006                  TRTRP:  RTT

                                ;ENTERED WITH SYSTEM TRAP CALL (HLT)
                                ;PRINT OUT THE ERROR PC+2, STATUS REGISTER, AND LOCATION IN BACKGROUND
016072  005767  000162          PRINT:  TST     PRTON           ;CHECK PRINT ON FLAG
016076  001401                          BEQ     .+4
016100  000002                          RTI                     ;IF ANOTHER HALT IS BEING PRINTED, SKIP THIS ONE
016102  005267  000152                  INC     PRTON
016106  012767  000340  161662          MOV     #340,PSR        ;SET PRIORITY TO 7
```

```
016114  736727  161450  020000          BIT     SR,#20000       ;TEST FOR INHIBIT PRINT OUT
016122  001401                          BEQ     .+4             ;BRANCH TO PRINT
016124  000444                          BR      CK              ;INHIBIT, CHECK FOR HALT
016126  012667  000122                  MOV     (6)+,SAVPC      ;PC OF FAILING ROUTINE
016132  012667  000120                  MOV     (6)+,SAVPSR     ;PSR OF ERROR CONDITION
016136  024646                          CMP     -(6),-(6)       ;RESTORE STACK
016140  012767  000200  161630          MOV     #200,PSR
016146  004767  000326                  JSR     %7,CRLF         ;OUTPUT CARRIAGE RETURN AND LINE FEED
016152  016767  000076  000260          MOV     SAVPC,PTEMP1    ;LOAD WITH FAILING PC+2
016160  004767  000076                  JSR     %7,PROCT        ;PRINT FAILING PC+2
016164  004767  000254                  JSR     %7,SPACE
016170  016767  000062  000242          MOV     SAVPSR,PTEMP1   ;LOAD PROCESSOR STATUS
016176  004767  000060                  JSR     %7,PROCT        ;PRINT PROCESSOR STATUS
016202  004767  000236                  JSR     %7,SPACE
016206  016767  162356  000224          MOV     CURBNK,PTEMP1
016214  004767  000042                  JSR     %7,PROCT
016220  004767  000220                  JSR     %7,SPACE
016224  016767  176416  000206          MOV     RETURN,PTEMP1
016232  004767  000024                  JSR     %7,PROCT
016236  005767  161326          CK:     TST     SR              ;CHECK SR FOR HALT SWITCH
016242  100001                          BPL     .+4             ;BRANCH IF NOT SET
016244  000000                          HALT                    ;HALT ON ERROR UP
016246  005067  000006                  CLR     PRTON           ;ROUTINE DONE - CLEAR FLAG
016252  000002                          RTI                     ;RETURN TO MAIN LINE
016254  000000          SAVPC:  0
016256  000000          SAVPSR: 0
016260  000000          PRTON:  0

                                ;SUBROUTINE TO PRINT OUT OCTAL NUMBER/
016262  012727  000006  016266  PROCT:  MOV     #6,#PTEMP3      ;CLEAR R4 FOR COUNTING CHARACTERS OUTPUT
        016266                          PTEMP3=.-2
016270  005067  000142                  CLR     PRFLG           ;INITIALIZE CARRY FLAG FOR ROTATES
016274  012767  000260  000140          MOV     #260,PTEMP2     ;SETUP R3
016302  005767  000132                  TST     PTEMP1          ;CHECK BIT 15 OF NUMBER
016306  100002                          BPL     .+6             ;BRANCH IF ZERO
016310  005267  000126                  INC     PTEMP2          ;INCREMENT R3 IF ONE
016314  006167  000120                  ROL     PTEMP1          ;ROTATE LEFT MOST OCTAL TO RIGHT END
016320  006167  000114                  ROL     PTEMP1
016324  005567  000106                  ADC     PRFLG           ;STORE CARRY
016330  105777  162054          P.WAIT: TSTB    @TCSR           ;WAIT FOR TTY READY
016334  100375                          BPL     P.WAIT
016336  016777  000100  162046          MOV     PTEMP2,@TDBR    ;OUTPUT NEXT CHARACTER
016344  005367  177716                  DEC     PTEMP3          ;COUNT
016350  001001                          BNE     P.CNT1          ;BRANCH IF NOT DONE
016352  000207                          RTS     %7              ;BRANCH IF NOT DONE
016354  000241          P.CNT1: CLC                             ;CLEAR CARRY
016356  005767  000054                  TST     PRFLG           ;CHECK FOR PREVIOUS CARRY
016362  001403                          BEQ     .+10            ;BRANCH IF PREVIOUSLY ZERO
016364  005067  000046                  CLR     PRFLG           ;INITIALIZE FLAG
016370  000261                          SEC                     ;SET CARRY
016372  006167  000042                  ROL     PTEMP1          ;ROTATE NEXT CHARACTER INTO RIGHT END OF REGISTER
016376  006167  000036                  ROL     PTEMP1
016402  006167  000032                  ROL     PTEMP1
016406  005567  000024                  ADC     PRFLG           ;STORE CARRY
```

```
016412  016767  000022  000022          MOV     PTEMP1,PTEMP2   ;LOAD DATA INTO R3
016420  042767  177770  000014          BIC     #177770,PTEMP2  ;CLEAR ALL BUT LOWEST OCTAL DIGIT
016426  052767  000260  000006          BIS     #260,PTEMP2     ;SET TO ASCII EQUIVALENT
016434  000735                          BR      P,WAIT          ;LOOP
016436  000000                  PRFLG:  0
016440  000000                  PTEMP1: 0                       ;CONTAINS VALUE TO BE OUTPUT
016442  000000                  PTEMP2: 0                       ;SCRATCH

                                ;SUBROUTINE TO ISSUE SPACE/
016444  105777  161740          SPACE:  TSTB    @TCSR           ;WAIT FOR TTY READY
016450  100375                  BPL     .-4
016452  012777  000240  161732          MOV     #240,@TDBR      ;OUTPUT A SPACE
016460  000207                  RTS     X7                      ;RETURN

                                ;BELL ON PASS COMPLETE
016462  105777  161722          BELL:   TSTB    @TCSR
016466  100375                  BPL     .-4
016470  012777  000207  161714          MOV     #207,@TDBR
016476  000207                  RTS     X7

                                ;SUBROUTINE TO OUTPUT CARRIAGE RETURN AND LINEFEED/
016500  105777  161704          CRLF:   TSTB    @TCSR           ;WAIT FOR TTY READY
016504  100375                  BPL     .-4
016506  012777  000215  161676          MOV     #215,@TDBR      ;OUTPUT CARRIAGE RETURN
016514  105777  161670          WAIT:   TSTB    @TCSR           ;WAIT FOR TTY READY
016520  100375                  BPL     .-4
016522  012777  000212  161662          MOV     #212,@TDBR      ;OUTPUT LINEFEED
016530  000207                  RTS     X7                      ;RETURN

                                ;ENTER HERE ON POWER FAIL/
016532  013746  000024          PFAIL:  MOV     @#24,-(6)
016536  010667  000010                  MOV     X6,SAVR6                ;STORE STACK POSITION
016542  012737  016554  000024          MOV     #RESTRT,@#24
016550  000000                  HALT                            ;HALT ON POWER DOWN NORMAL
016552  000000                  SAVR6:  0                       ;STACK IS SAVED HERE
016554  016706  177772          RESTRT: MOV     SAVR6,X6        ;RESTORE STACK WHEN POWERING UP
016560  012637  000024                  MOV     (6)+,@#24
016564  022626                  CMP     (SP)+,(SP)+             ;RESTORE STACK
016566  104006                  HLT                             ;POWER FAIL OCCURRED
016570  000167  162022                  JMP     RSTRT
                                                                ;RETURN TO MAIN LINE

016574  000207          USER:   RTS     X7                      ;OVERLAY USER ROUTINE HERE IF 4KW
                                                                ;USE BANK1 IF 8KW
        017760                  .=17760
017760  000000          KSTACK: 0
        000001                  .END
```

```
A          014660          ADSUB      014040          ARIEND     014102          ARITST     014014
B          014652          BEGIN      005376          BEGINX     005344          BELL       016462
BNKLP      016052          BNKSTR     000600          BNKT       015602          BNKTST     015466
BNKT1      015632          C          014672          CK         016236          CLINCT   = 003174
CMP1       013340          CMP2       013362          COMPAR     013334          COREPT     000610
CORMOV     015752          CRLF       016500          CURBNK     000570          CURPAR     000574
CURPAT   = 003104          CURPDR     000576          D          014712          DATA2    = 002712
DET1       002274          DET2       002376          DET3       002674          DET4       002326
DO       = 000001          DONE       014506          DSW1       015154          DSW2       015170
DSW3       015204          DSW4       015220          DSW5       015234          EIGHT      002344
EMTAB      015016          EMTSRV     014756          EOB      = 104010          EOBSRV     015030
EOB1       015046          EOB1A      015124          EOB1B      015250          EOB1C      015256
EOB2       015264          EOB3       015704          EOB6       016056          EPC        015014
ERROK      003462          ERRSET     003456          FENDZ      003536          FEND1      003560
HLT      = 104006          ICOUNT     014642          IE       = 000500          IPDREN     000544
IPDRTA     000536          IRF        003416          IRK        003322          KPAR0      000526
KPAR1      000530          KPAR2      000532          KPAR7      000534          KPDR0      000516
KPDR1      000520          KPDR2      000522          KPDR7      000524          KSTACK     017760
KWLST      000424          KWLVC      000422          LKCSR      000426          LK1        002760
LK2        003000          LK3        003002          LK4        003020          LLIMIT     003524
LOGIC      015550          LOGICA     015570          LPCSR      000430          LPDBR      000432
LPINTR     003124          LPST       000436          LPVC       000434          LP1        003100
LP2        003106          LP3        003214          LP4        003224          LP5        003262
LP6        003162          LP7        003160          MAIN       002242          MAP1       001102
MAP2       001172          MEMUT      000612          MEM0       000604          MEM1       000606
MODE       002172          MOD12      002630          MOD16      002614          MOD20      002600
MOD24      002564          MOD4       002660          MOD8       002644          MOVE       002410
MOVEPT     001142          N        = 000001          NODEV      002260          NOP      = 000240
NRALL      014714          NRLOOP     014726          NUMA       014100          NXTBNK     015432
NXTSEG     015362          OLDBNK     000572          PATCH1   = 000000          PATCH2   = 000000
PATCH3   = 000000          PC       =%000007          PFAIL      016532          PRFLG      016436
PRINT      016072          PROCT      016262          PRTON      016260          PSR      = 177776
PTEMP1     016440          PTEMP2     016442          PTEMP3   = 016266          P.CNT1     016354
P.WAIT     016330          R        = 004000          RB       = 000002          RD       = 000004
REF      = 013526          REFF       013526          RENDZ      003734          REND1      003764
RESTRT     016554          RETURN     014646          RFCAR      000446          RFCSR      000450
RFCSRH     000452          RFDAE      000440          RFDAR      000442          RFFUNC     003522
RFST       000456          RFSTAR     003366          RFVC       000454          RFWC       000444
RF1        003372          RKBAR      000466          RKCSR      000470          RKCSRH     000472
RKDAE      000462          RKDAH      000460          RKFUNC     003364          RKST       000476
RKSTAR     003272          RKVC       000474          RKWC       000464          RK1        003276
ROTALL     013416          ROTBE      013566          ROTBO      013676          ROTEN1     014000
RSTRT      000616          R0       =%000000          R1       =%000001          R2       =%000002
R3       =%000003          R4       =%000004          R5       =%000005          R6       =%000006
SAVPC      016254          SAVPSR     016256          SAVR6      016552          SCOPE    = 104400
SCOPEB     014622          SCOPEC     014550          SCOPEF     014644          SCOPEG     014630
SEGM1      002040          SETEX      002020          SETSEG     002164          SETUSE     002070
SOLPAT   = 003102          SP       =%000006          SPACE      016444          SR       = 177570
SREG1      000546          SREG2      000550          SR0        000500          START      000640
STARTX     000754          START1     000772          STRT28     002474          STRT4      002430
ST3        001354          ST4        001440          ST5        001502          ST6        001566
ST7        001624          ST8        001720          SUBR1      014512          SUBR2      014514
SUBR3      014520          SUBR4      014526          SUBR5      014534          SUBR6      014542
SWABA      014442          TBANK      000614          TCBA       000562          TCCM       000552
```

DBKTGA,SRC SYMBOL TABLE

```
TCDT    000556          TCEXPE  003534          TCFIRS  003530          TCF1    003610
TCF1A   003602          TCF2    003642          TCF3    003656          TCF4    003724
TCIV    000564          TCLAST  003532          TCOM    014114          TCOM2   014162
TCOM3   014230          TCRBK   004214          TCRBUF  004344          TCRB1   004252
TCR1    004062          TCR1A   004110          TCR2    004124          TCR3    004140
TCR4    004206          TCSR    = 000410        TCST    000554          TCSTA   000564
TCWBK   004002          TCWBUF  004344          TCWB1   004034          TCWC    000560
TC1     004300          TC2     004320          TDBR    = 000412        TEMP    014702
TEST    013530          TIME    003076          TJSR1   012712          TJSR2   012714
TJSR3   012726          TMEMEX  001132          TRCSR   000406          TRPB    000602
TRTRP   016070          TSCOMB  014272          TSROT   013374          TSROT2  013540
TSRT2A  013532          TTCSR   000410          TTDBR   000412          TTPST   000416
TTPVC   000414          TTSAV   000420          TYOUT   002710          TYOUTR  002724
TYOUT1  002714          UBUFF   000400          UPAR0   000510          UPAR1   000512
UPAR7   000514          UPDR0   000502          UPDR1   000504          UPDR7   000506
USEALL  002112          USER    016574          WAIT4   014302          WD      = 000014
WORDCT  003526          XFEND   003732          XFER12  002542          XFER16  002532
XFER20  002522          XFER24  002512          XFER28  002502          XFER8   002552
.       = 017762
```

ERRORS DETECTED: 0

```
DBKTGA  MACY11,620  8-NOV-72  09:15  PAGE 15-56
DBKTGA,SRC

 *DBKTGA,DBKTGA=DBKTGA,SRC
 RUN-TIME:  13 25 0 SECONDS
 CORE USED:  4K
```

IDENTIFICATION

| | |
|---|---|
| PRODUCT CODE: | MAINDEC-11-DBKTF-A-D |
| PRODUCT NAME: | MEMORY MANAGEMENT ABORT TESTS |
| DATE CREATED: | SEPTEMBER 1, 1972 |
| MAINTAINER: | DIAGNOSTIC GROUP |
| AUTHOR: | ROBERT WHITTON |

;COPYRIGHT 1972, DIGITAL EQUIPMENT CORP., MAYNARD, MASS.

ADVANCE COPY
This document subject to change without notice.

COPYRIGHT © 1972
DIGITAL EQUIPMENT CORPORATION

1,0 ABSTRACT

PROGRAM DBKTF TESTS THE MEMORY MANAGEMENT ABORT LOGIC, THE PROGRAM IS WRITTEN TO CAUSE A MEMORY MANAGEMENT ABORT AT EVERY PDP11/40 MICRO STATE WHERE A MEMORY REFERENCE IS INITIATED, ABORTS ARE IN ALL CASES TRAPPED TO THE KERNEL, HOWEVER, THE INSTRUCTIONS CAUSING THE ABORT ARE EXECUTED IN BOTH MODES (KERNEL AND USER).

2,0 REQUIREMENTS

2,1 EQUIPMENT

PDP-11/40 WITH KT11-D (MEM, MGMT) INSTALLED

2,2 STORAGE

PROGRAM STORAGE - THE ROUTINE USES MEMORY 0-17777

2,3 PRELIMINARY PROGRAMS

TESTS DBKTA-DBKTD

3,0 LOADING AND STARTING PROCEDURE

LOAD PROGRAM INTO MEMORY USING ABS LOADER
LOAD ADDRESS 200
PRESS START.
THE PROGRAM WILL LOOP AND RING BELL ON COMPLETION.

4,0 SWITCH SETTINGS

## 5,0 SUBROUTINE ABSTRACTS

### 5,1 HLT

THE HLT (HALT) INSTRUCTION IS EXECUTED WHEN AN ERROR IS DETECTED NOTE THAT THE HLT (HALT) INSTRUCTION TRAPS TO LOC 4 IN USER MODE, IF A HLT (HALT) INSTRUCTION IS EXECUTED IN THESE MODES THE TRAP IS TAKEN AND THE PROGRAM HALTS AT LOCATION 176 IN KERNEL MODE, PRESSING CONTINUE RESTARTS THE TEST, NOTE! THE USER STACK POINTER IS NOT AFFECTED, TO DETRMINE WHICH TEST THE PROGRAM WAS EXECUTING WHEN THE HLT OCCURRED REFER TO R1 WHOSE CONTENTS ARE THE LAST TEST SUCCESSFULLY EXECUTED AND ALSO THE KERNEL STACK THE TOP WORD OF WHICH IS THE VIRTUAL PC OF THE HLT INSTRUCTION +2,

### 5,2 SCOPE

THE SCOPE (EMT) SERVICE ROUTINE STORES IN R1 THE PC OF THE LAST TEST SUCCESSFULLY EXECUTED AND MAY BE USED AS AN AID IN DEBUGGING IF THE PROGRAM 'BOMBS' BECAUSE OF A HARDWARE FAILURE, A BRANCH INSTRUCTION MAY BE INSERTED AT THE SCOPE LOCATION TO THE PREVIOUS SCOPE (EMT) INSTRUCTION TO CONTINOUSLY LOOP A TEST, ADDITIONALLY THE SCOPE ROUTINE SETS ALL STACK POINTERS TO THEIR INITIAL SETTINGS (SEE SEC 8,2) AND ENTERS EACH TEST IN KERNEL MODE, PREVIOUS KERNEL MODE,

## 6,0 ERRORS

THE TEST HALTS WHEN AN ERROR IS DETECTED AND DISPLAYS THE PC+2 OF THE HLT (HALT) INSTRUCTION IN THE ADDRESS LIGHTS,

### 6,1 ERROR RECOVERY

PRESS CONTINUE OR RESTART AT 200 OR PREVIOUS SCOPE,

### 6,2 ERROR LOOPING

TO LOOP ON AN ERROR REPLACE THE HLT INSTRUCTION WITH A BRANCH BACK TO THE PREVIOUS SCOPE, NOTE! IF THE ERROR IS INTERMITTENT THE TEST WILL DROP THROUGH THE HLT AND CONTINUE TO THE NEXT TEST, TO CONTINUOSLY LOOP THE TEST REPLACE THE BEQ ,+4 PRECEEDING THE HLT WITH THE BRANCH,

7,0 RESTRICTIONS

7,1 STARTING RESTRICTION

NONE

7,2 OPERATIONAL RESTRICTION

NONE

8,0 MISCELLANEOUS

IF THE PROGRAM HALTS IN THE TRAP INTERRUPT VECTOR AREA (0-1000) EXAMINE REGISTER 6 (THE KERNEL STACK PTR), REGISTER 6 CONTAINS THE ADDRESS WHERE THE PC OF THE INSTRUCTION THAT CAUSED THE TRAP IS STORED, EXAMINE ALSO R1 (R1 SPECIFIES THE LAST TEST SUCCESSFULLY COMPLETED)

8,2 STACK POINTER

THE STACK POINTERS ARE INITIALLY SET TO THE FOLLOWING VALUES
KERNEL =1060
USER = 600
AND ARE RESET TO THESE VALUES AT THE START OF EACH SUBTEST (BY SCOPE).

8,3 PASS COUNT

1000(8) PASSES ARE REQUIRED FOR COMPLETION OF THIS PROGRAM, AT WHICH TIME THE BELL WILL RING AT THE TTY.

8,4 DEBUGGING TIPS

WHEN THE FAILING SUBTEST HAS BEEN ISOLATED, REPLACE THE FIRST WORD OF THE INSTRUCTION PRECEEDING THE INSTRUCTION THAT CAUSES THE ABORT WITH A BR SELF (000777), AND RESTART THE PROGRAM. WHEN THE PROGRAM EXECUTES THE BR SELF STOP THE PROGRAM USING SINGLE INSTRUCTION, RESTORE THE INSTRUCTION, AND USING THE MAINTENANCE CARD SINGLE STEP THE PROGRAM THROUGH EACH MICRO STATE OBSERVING THE FLOW IN THE DATA/ADDRESS LIGHTS. THIS PRACTICE HAS BEEN FOUND TO BE SUCCESSFUL IN FINDING MOST MEMORY MANAGEMENT ERRORS.

8,5 MEMORY MANAGEMENT MEMORY MAP

THE MAPPING OF THE MEM MGMT REGISTERS IS DONE AT THE BEGINNING OF THE PROGRAM BEFORE ANY TESTING IS STARTED. THE USER SHOULD ACQUAINT HIMSELF WITH THE MEMORY MANAGEMENT MAP BEFORE USING THIS PROGRAM.

```
1                                       .LIST   ME
                                        .NLIST  SEQ
                                        .ABS
                                        .TITLE DBKTFA
                                ;MEMORY MANAGEMENTGEMENT ABORT TEST, THIS PROGRAM TESTS MEMORY MGMT ABORT ERRORS

                                ;GENERAL REGISTER ASSIGNMENTS
000000                                  R0=%0
000001                                  R1=%1
000002                                  R2=%2
000003                                  R3=%3
000004                                  R4=%4
000005                                  R5=%5
000006                                  SP=%6
000007                                  PC=%7

                                ;STACK POINTER REGISTERS
000006                                  KSP=%6                  ;KERNEL STACK POINTER
000006                                  USP=%6                  ;USER STACK POINTER

                                ;STATUS REGISTER BIT ASSIGNMENTS
000001                                  C=1
000002                                  V=2
000004                                  Z=4
000010                                  N=10
000020                                  T=20                    ;'T' BIT
000340                                  PRTY7=340               ;PRIORITY LEVEL 7
000200                                  PRTY4=200               ;PRIORITY LEVEL 4
000000                                  KM=000000               ;KERNEL MODE
140000                                  UM=140000               ;USER MODE
000000                                  PKM=000000              ;PREVIOUS KERNEL MODE
030000                                  PUM=030000              ;PREVIOUS USER MODE

                                ;VECTOR ADDRESSES
000010                                  ERRVEC=10               ;ADDRESS OF ERROR VECTOR
000014                                  TBITVEC=14              ;ADDRESS OF 'T' BIT TRAP VECTOR
000020                                  IOTVEC=20               ;ADDRESS OF IOT TRAP VECTOR
000024                                  PFVEC=24                ;ADDRESS OF POWER FAIL TRAP VECTOR
000030                                  EMTVEC=30               ;ADDRESS OF EMT VECTOR
000034                                  TRAPVEC=34              ;ADDRESS OF TRAP VECTOR
000064                                  TPVEC=64                ;ADDRESS OF TTY PRINTER INTERRUPT VECTOR
000244                                  FPVEC=244               ;ADDRESS OF FLOATING POINT INT, VECTOR
000250                                  MMVEC=250               ;ADDRESS OF MEMORY MGMT ERROR TRAP VECTOR

                                ;REGISTER ADDRESSES
177776                                  PSW=177776              ;ADDRESS OF STATUS REGISTER
177560                                  TKS=177560              ;ADDRESS OF KEYBOARD CSR
177562                                  TKB=177562              ;ADDRESS OF KEYBOARD BUFFER
177564                                  TPS=177564              ;ADDRESS OF TELEPRINTER CSR
177566                                  TPB=177566              ;ADDRESS OF TELEPRINTER BUFFER
177570                                  SWR=177570              ;ADDRESS OF CONSOL SWITCH REGISTER

                                ;INITIAL STACK POINTER SETTINGS
001100                                  KPTR=1100               ;BOTTOM OF KERNEL STACK
```

```
000600                          UPTR=600                ;USER STACK SETTING

                        ;MISCELLANEOUS BIT ASSIGNMENTS
100000                          BIT15=100000
040000                          BIT14=40000
020000                          BIT13=20000
000400                          BIT8=400
000100                          BIT6=100

                        ;MEMORY MANAGEMENT REGISTER SR0 BIT ASSIGNMENTS
000001                          ENMM=1                  ;ENABLE MEMORY MANAGEMENT
000000                          VS0=0
000002                          VS1=2
000004                          VS2=4
000006                          VS3=6
000010                          VS4=10
000012                          VS5=12
000014                          VS6=14
000016                          VS7=16
000000                          IS=00
000140                          UPG=140
000000                          KPG=000
000400                          DM=400                  ;DESTINATION MODE
020000                          AVA=20000               ;ACCESS VIOLATION ABORT
040000                          PLA=40000               ;PAGE LENGTH ABORT
100000                          NRA=100000              ;NON-RESIDENT ABORT

                        ;PAGE DESCRIPTOR REGISTER (PDR) BIT ASSSIGNMENTS
000010                          ED=10                   ;EXPANSION DIRECTION BIT IN PDR
000000                          UP=0                    ;EXPAND UP
000010                          DWN=10                  ;EXPAND DOWN
000100                          W=100                   ;'W' BIT IN PDR

                        ;MEMORY MANAGEMENT REGISTER ADDRESS ASSIGNMENTS
177572                          SR0=177572              ;ADDRESS OF MEMORY MGMT REGISTER SR0
177574                          SR1=177574              ;    "    "      "          "    SR1
177576                          SR2=177576              ;    "    "      "          "    SR2

177600                          UIPDR0=177600           ;ADDRESS OF USER 'I' PDR'S
177602                          UIPDR1=177602
177604                          UIPDR2=177604
177606                          UIPDR3=177606
177610                          UIPDR4=177610
177612                          UIPDR5=177612
177614                          UIPDR6=177614
177616                          UIPDR7=177616

177640                          UIPAR0=177640
177642                          UIPAR1=177642
177644                          UIPAR2=177644
177646                          UIPAR3=177646
177650                          UIPAR4=177650
177652                          UIPAR5=177652
177654                          UIPAR6=177654
```

```
                177656                          UIPAR7=177656

                172300                          KIPDR0=172300
                172302                          KIPDR1=172302
                172304                          KIPDR2=172304
                172306                          KIPDR3=172306
                172310                          KIPDR4=172310
                172312                          KIPDR5=172312
                172314                          KIPDR6=172314
                172316                          KIPDR7=172316

                172340                          KIPAR0=172340
                172342                          KIPAR1=172342
                172344                          KIPAR2=172344
                172346                          KIPAR3=172346
                172350                          KIPAR4=172350
                172352                          KIPAR5=172352
                172354                          KIPAR6=172354
                172356                          KIPAR7=172356

                                        ;ACCESS CONTROL FIELD DEFINITIONS (IN PDR)
                000000                          NR0=0                   ;NON-RESIDENT ABORT ALL REFS.
                000002                          RO0=2                   ;READ,ABORT ON WRITE
                000004                          RWT=4                   ;TRAP ON READ & WRITE
                000006                          RW=6                    ;READ & WRITE

                                        ;INSTRUCTION EQUATES
                000000                          HLT=HALT
                104000                          SCOPE=EMT               ;SCOPE IS AN EMT TRAP


                                        ;VIRTUAL ADDRESSES
                016700                          KI0=16700
                140000                          KI6=140000
                120000                          UI5=120000
                100000                          UI4=100000
                040000                          UI2=40000
                020000                          UI1=20000
                060000                          UI3=60000
                                        ;CORRESPONDING PHYSICAL ADDRESSES
                016600                          PKI0=16600
                016700                          PKI6=16700
                017200                          PUI5=17200
                017300                          PUI4=17300
                017400                          PUI3=17400
                017000                          PUI2=17000
                017100                          PUI1=17100
                                                .LIST   ME
                                                .NLIST  MC,MD
                                        ;FILL TRAP AND INTERRUPT VECTOR AREA WITH
                                                ;,.+2
                                                ;HALT
                                        ;UNEXPECTED TRAPS/INTERRUPTS WILL HALT AT VECTOR ADDRESS .+2
                                        ;AND DISPLAY VECTOR ADDRESS+4 NOTE! LISTING DOES NOT SHOW LOADING THE
```

```
                                ;VECTOR AREA,

                                        .NLIST  MC
        000010                          .=ERRVEC
000010  000400                          .WORD   SHLT
        000030                          .=EMTVEC
000030  000442                          .WORD   SCOPEA

        000176                          .=176
000176  000000                          HALT                    ;ERROR! TO IDENTIFY WHICH TEST FAILED
                                ;EXAMINE R1, THE CONTENTS OF WHICH IS THE PC OF THE PRESENT TEST
                                ;THE TOP WORD ON THE KERNEL STACK CONTAINS THE VIRTUAL
                                ;ADDRESS OF THE HLT INSTRUCTION IN THE TEST THAT FAILED.
        000200                          .=200
000200  000167  001006                  JMP     START           ;GO START TEST

        000400                          .=400
                                ;USER HLT (HALT) TRAP SERVICE ROUTINE
000400  042737  000001  177572  SHLT:   BIC     #1,@#SR0        ;TURN MEM MGMT OFF
000406  042737  140000  177776          BIC     #140000,@#PSW   ;RETURN TO KERNEL
000414  162716  000002                  SUB     #2,(KSP)        ;POINT PC TO TRAPPING INST.
000420  005776  000000                  TST     @(KSP)          ;WAS IT A HLT (HALT)
000424  001404                          BEQ     SHLTA
000426  062716  000002                  ADD     #2,(KSP)        ;RESTORE PC TO TRAPPING INST.
000432  000137  000012                  JMP     @#ERRVEC+2      ;GO HALT AT 6
000436  000137  000176          SHLTA:  JMP     @#176           ;GO HALT AT ADDRESS 176

                                ;SCOPE (EMT) SERVICE ROUTINE
000442  005037  177572          SCOPEA: CLR     @#SR0           ;DISABLE MEMORY MGMT
000446  011601                          MOV     (KSP),R1        ;SAVE PC IN R1
000450  012706  001100                  MOV     #KPTR,KSP       ;SET KERNEL STACK PTR
000454  005046                          CLR     -(KSP)          ;SET UP FOR KERNEL MODE ON RETURN
000456  010146                          MOV     R1,-(KSP)       ;RETURN IN LINE
000460  012746  000600                  MOV     #UPTR,-(KSP)    ;USER STACK PTR ON KERNEL STACK
000464  012737  030000  177776          MOV     #PUM,@#PSW      ;PREVIOUS USER MODE
000472  106606                          MTPD    USP             ;SET USER STACK PTR
000474  001400                          BEQ     SCOPEX
000476  000006          SCOPEX: RTT                             ;RETURN TO NEXT TEST IN KERNEL MODE
                                                                ;WITH ALL STACK PTRS SET UP

        001200                          .=1200
                                ;TAGS
001200  000000          ICNT:   0                               ;CONTAINS PASS COUNT
001202  000000          SR0T:   0                               ;CONTAINS SR0 CONTENTS ON ERROR
        001204                  TEMP=.
        001212                  .=.+6
```

```
                                ;START MEMORY MANAGEMENT TEST,
001212  000240                 START:  NOP
001214  005067  177760                 CLR     ICNT                    ;CLEAR PASS COUNT
001220  012706  001100         BEGIN:  MOV     #KPTR,KSP               ;SET KERNEL STACK PTR
001224  104000                         SCOPE                           ;SCOPE SETS ALL STACK PTRS
001226  005037  000252                 CLR     @#MMVEC+2               ;KERNEL MODE ON ABORT

                                ;ROUTINE TO CLEAR MEMORY MANAGEMENT REGISTERS,
001232  000240                 MM0:    NOP
001234  005067  176332                 CLR     SR0
001240  012702  177600                 MOV     #UIPDR0,R2
001244  012703  000010                 MOV     #8.,R3
001250  005022                         CLR     (R2)+
001252  077302                         SOB     R3,.-2
001254  012702  177640                 MOV     #UIPAR0,R2
001260  012703  000010                 MOV     #8.,R3
001264  005022                         CLR     (R2)+
001266  077302                         SOB     R3,.-2
001270  012702  172300                 MOV     #KIPDR0,R2
001274  012703  000010                 MOV     #8.,R3
001300  005022                         CLR     (R2)+
001302  077302                         SOB     R3,.-2
001304  012702  172340                 MOV     #KIPAR0,R2
001310  012703  000010                 MOV     #8.,R3
001314  005022                         CLR     (R2)+
001316  077302                         SOB     R3,.-2

001320  012737  073006  172300 MMK:    MOV     #73006,@#KIPDR0 ;RW, UP 167 BLOCKS
001326  012737  000006  172314         MOV     #6,@#KIPDR6     ;RW, UP 1 BLOCK
001334  012737  077406  172316         MOV     #77406,@#KIPDR7 ;RW, UP 200 BLOCKS
001342  012737  073006  177600         MOV     #73006,@#UIPDR0 ;RW, UP 167 BLOCKS
001350  012737  000006  177604         MOV     #6,@#UIPDR2     ;RW, UP 1 BLOCK
001356  012737  000006  177602         MOV     #6,@#UIPDR1     ;RW, UP 1 BLOCK
001364  012737  000006  177610         MOV     #6,@#UIPDR4     ;RW, UP 1 BLOCK
001372  012737  000006  177612         MOV     #6,@#UIPDR5     ;RW, UP 1 BLOCK

001400  005067  170734                 CLR     KIPAR0          ;VA=PA=0000-16677
001404  012767  000167  170742         MOV     #167,KIPAR6     ;VA=140000-140077/PA=16700-16777
001412  012767  007600  170736         MOV     #7600,KIPAR7    ;VA=160000-177776,PA=760000-777776
001420  005067  176214                 CLR     UIPAR0          ;VA=PA=0-16677
001424  012767  000170  176212         MOV     #170,UIPAR2     ;VA=40000-40077/PA=17000-17077
001432  012767  000171  176202         MOV     #171,UIPAR1     ;VA=20000-20077/PA=17100-17177
001440  012767  000172  176204         MOV     #172,UIPAR5     ;VA=120000-120077/PA=17200-17277
001446  012767  000173  176174         MOV     #173,UIPAR4     ;VA=100000-100077/PA=17300-17377
```

```
                                ;CHECK ABORT AT SRC00
                                ;ABORTS WHEN SOURCE OPERAND IS FETCHED

                                ;SOURCE MODE=1
001454                          T0:
001454  012737  001510  000250          MOV     #T0C,@#MMVEC    ;LOAD MEM MGMT ERROR VECTOR
001462  005067  176564                  CLR     MMVEC+2
001466  012703  016700                  MOV     #K:0,R3
001472  010302                          MOV     R3,R2
001474  005013                          CLR     (R3)
001476  005237  177572                  INC     @#SR0           ;ENABLE MEMORY MGMT
001502  000277                          SCC
001504  011302                  T0A:    MOV     (R3),R2         ;MEM MGMT LENGTH ABORT AT SRC00
001506  000000                  T0B:    HLT                     ;ERROR: DID NOT ABORT
001510  022706  001074          T0C:    CMP     #KPTR+4,KSP     ;CHECK STACK PTR
001514  001401                          BEQ     .+4
001516  000000                          HLT
001520  022766  000017  000002          CMP     #17,2(KSP)      ;CHECK THAT CORRECT STATUS
001526  001401                          BEQ     .+4             ;WAS SAVED ON THE STACK
001530  000000                          HLT                     ;ERROR: INCORRECT STATUS
001532  022767  040001  176032          CMP     #PLA+1,SR0      ;CHECK SR0 (ABORT CONDITIONS
001540  001401                          BEQ     .+4             ;& FAILING PAGE #)
001542  000000                          HLT                     ;ERROR: INCORRECT ABORT CONDITIONS OR PAGE IDENT
001544  022767  001504  176024          CMP     #T0A,SR2        ;CHECK CONTENTS OF SR2
001552  001401                          BEQ     .+4             ;(PC OF ABORTED INSTRUCTION)
001554  000000                          HLT                     ;ERROR: INCORRECT PC IN SR2
001556  020203                          CMP     R2,R3           ;CHECK THAT INSTRUCTIONS AS ABORTED
001560  001401                          BEQ     .+4
001562  000000                          HLT                     ;ERROR!
001564  104000                          SCOPE                   ;SCOPE STORES PC IN R1 & SETS ALL STACK PTRS

                                ;CHECK ABORT AT SRC01
                                ;ABORTS WHEN SOURCE OPERAND IS FETCHED
                                ;SOURCE MODE=2,BYTE INSTRUCTION
001566  012737  001614  000250          MOV     #T1C,@#MMVEC    ;LOAD MEM MGMT ERROR VECTOR
001574  012702  016700                  MOV     #K:0,R2
001600  010204                          MOV     R2,R4
001602  005012                          CLR     (R2)
001604  005237  177572                  INC     @#SR0           ;ENABLE MEMORY MGMT
001610  122202                  T1A:    CMPB    (R2)+,R2        ;SEG LENGTH ABORT AT SRC01
001612  000000                  T1B:    HLT                     ;ERROR: DID NOT ABORT
001614  022706  001074          T1C:    CMP     #KPTR+4,KSP     ;CHECK STACK PTR
001620  001401                          BEQ     .+4
001622  000000                          HLT
001624  022767  040001  175740          CMP     #PLA+1,SR0      ;CHECK SR0 (ABORT CONDITIONS
001632  001401                          BEQ     .+4             ;& FAILING PAGE #)
001634  000000                          HLT                     ;ERROR: INCORRECT ABORT CONDITIONS OR PAGE IDENT
001636  022767  001610  175732          CMP     #T1A,SR2        ;CHECK CONTENTS OF SR2
001644  001401                          BEQ     .+4             ;(PC OF ABORTED INSTRUCTION)
001646  000000                          HLT                     ;ERROR: INCORRECT PC IN SR2
                                ;CHECK THAT REGISTER INCREMENTED PROPERLY
001650  022702  016701                  CMP     #K:0+1,R2
001654  001401                          BEQ     .+4
```

```
001656  000000                          HLT                     ;ERROR!
001660  104000                          SCOPE                   ;SCOPE STORES PC IN R1 & SETS ALL STACK PTRS

                                ;CHECK ABORT AT SRC13
                                ;ABORTS WHEN ADDRESS OF SOURCE OPERAND IS FETCHED
                                ;SOURCE MODE=3
001662  012737  001710  000250          MOV     #T2C,@#MMVEC    ;LOAD MEM MGMT ERROR VECTOR
001670  012705  016700                  MOV     #KI0,R5
001674  010504                          MOV     R5,R4
001676  005237  177572                  INC     @#SR0           ;ENABLE MEMORY MGMT
001702  000277                          SCC                     ;PRESET CC'S
001704  153504                  T2A:    BISB    @(R5)+,R4       ;NON-RES ABORT AT S13,10
001706  000000                  T2B:    HLT                     ;ERROR! FAILED TO ABORT
001710                          T2C:
001710  022766  000017  000002          CMP     #17,2(KSP)      ;CHECK THAT CORRECT STATUS
001716  001401                          BEQ     .+4             ;WAS SAVED ON THE STACK
001720  000000                          HLT                     ;ERROR! INCORRECT STATUS
001722  022767  040001  175642          CMP     #PLA+1,SR0      ;CHECK SR0 (ABORT CONDITIONS
001730  001401                          BEQ     .+4             ;& FAILING PAGE #)
001732  000000                          HLT                     ;ERROR! INCORRECT ABORT CONDITIONS OR PAGE IDENT
001734  022767  001704  175634          CMP     #T2A,SR2        ;CHECK CONTENTS OF SR2
001742  001401                          BEQ     .+4             ;(PC OF ABORTED INSTRUCTION)
001744  000000                          HLT                     ;ERROR! INCORRECT PC IN SR2
001746  022705  016702                  CMP     #KI0+2,R5
001752  001401                          BEQ     .+4
001754  000000                          HLT                     ;ERROR! R5 DID NOT AUTO-INCREMENT
001756  104000                          SCOPE                   ;SCOPE STORES PC IN R1 & SETS ALL STACK PTRS

                                ;CHECK ABORT AT SRC02
                                ;ABORTS WHEN SOURCE OPERAND IS FETCHED
                                ;SOURCE MODE=4,USER MODE
001760  012737  002014  000250          MOV     #T3C,@#MMVEC    ;LOAD MEM MGMT ERROR VECTOR
001766  012767  170000  176002          MOV     #UM+PUM,PSW     ;USER MODE!!!, PREV USER MODE
001774  012702  100000                  MOV     #UI4,R2
002000  010203                          MOV     R2,R3
002002  005237  177572                  INC     @#SR0           ;ENABLE MEMORY MGMT
002006  000277                          SCC                     ;PRESET CC'S
002010  064203                  T3A:    ADD     -(R2),R3        ;NON-RESIDENT ABORT AT SRC02
002012  000000                  T3B:    HLT                     ;ERROR! FAILED TO ABORT
002014  022706  001074          T3C:    CMP     #KPTR+4,KSP     ;CHECK STACK PTR
002020  001401                          BEQ     .+4
002022  000000                          HLT
002024  022766  170017  000002          CMP     #UM+PUM+17,2(KSP)       ;CHECK THAT CORRECT STATUS
002032  001401                          BEQ     .+4             ;WAS SAVED ON THE STACK
002034  000000                          HLT                     ;ERROR! INCORRECT STATUS
002036  022767  140147  175526          CMP     #NRA+PLA+UPG+VS3+1,SR0  ;CHECK SR0 (ABORT CONDITIONS
002044  001401                          BEQ     .+4             ;& FAILING PAGE #)
002046  000000                          HLT                     ;ERROR! INCORRECT ABORT CONDITIONS OR PAGE IDENT
002050  022767  002010  175520          CMP     #T3A,SR2        ;CHECK CONTENTS OF SR2
002056  001401                          BEQ     .+4             ;(PC OF ABORTED INSTRUCTION)
002060  000000                          HLT                     ;ERROR! INCORRECT PC IN SR2
                                ;CHECK CONTENTS OF REFERENCED PAGE DESCRIPTOR REGISTER (UIPDR2)
002062  032767  000100  175514          BIT     #W,UIPDR2       ;CHECK CONTENTS OF REFERENCED PDR
002070  001401                          BEQ     .+4
```

```
002072  000000                          HLT                     ;ERROR!
002074  042767  000100  175502          BIC     #W,UIPDR2
002102  022702  077776                  CMP     #UI4-2,R2       ;CHECK THAT AUTO- DECREMENT TOOK PLACE
002106  001401                          BEQ     .+4
002110  000000                          HLT                     ;ERROR! R2 FAILED TO AUTO-DECREMENT
002112  022703  100000                  CMP     #UI4,R3         ;CHECK THAT R3 WAS NOT CHANGED
002116  001401                          BEQ     .+4
002120  000000                          HLT                     ;ERROR!
002122  104000                          SCOPE                   ;SCOPE STORES PC IN R1 & SETS ALL STACK PTRS

                                ;CHECK ABORT AT SRC13
                                ;ABORTS WHEN ADDRESS OF SOURCE OPERAND IS FETCHED
                                ;SOURCE MODE=5,USER MODE
002124  012737  002164  000250          MOV     #T4C,@#MMVEC    ;LOAD MEM MGMT ERROR VECTOR
002132  012767  170000  175636          MOV     #UM+PUM,PSW     ;USER MODE!!!,PREV USER MODE!!
002140  012704  120002                  MOV     #UI5+2,R4
002144  010405                          MOV     R4,R5
002146  012737  177777  017200          MOV     #-1,@#PUI5
002154  005237  177572                  INC     @#SR0           ;ENABLE MEMORY MGMT
002160  145405                  T4A!    BICB    @-(R4),R5       ;NON-RESIDENT ABORT AT SRC13
002162  000000                  T4B!    HLT                     ;ERROR! FAILED TO ABORT
002164  022706  001074          T4C!    CMP     #KPTR-4,KSP     ;CHECK STACK PTR
002170  001401                          BEQ     .+4
002172  000000                          HLT
002174  022767  140157  175370          CMP     #NRA+PLA+UPG+VS7+1,SR0  ;CHECK SR0 (ABORT CONDITIONS
002202  001401                          BEQ     .+4             ;& FAILING PAGE #)
002204  000000                          HLT                     ;ERROR! INCORRECT ABORT CONDITIONS OR PAGE IDENT
002206  022767  002160  175362          CMP     #T4A,SR2        ;CHECK CONTENTS OF SR2
002214  001401                          BEQ     .+4             ;(PC OF ABORTED INSTRUCTION)
002216  000000                          HLT                     ;ERROR! INCORRECT PC IN SR2
                                ;CHECK CONTENTS OF REFERENCED PAGE DESCRIPTOR REGISTER (UIPDR5)
002220  032767  000100  175364          BIT     #W,UIPDR5       ;CHECK CONTENTS OF REFERENCED PDR
002226  001401                          BEQ     .+4
002230  000000                          HLT                     ;ERROR!
002232  042767  000100  175352          BIC     #W,UIPDR5
002240  022704  120000                  CMP     #UI5,R4 ;CHECK AUTO-DECREMENT
002244  001401                          BEQ     .+4
002246  000000                          HLT                     ;ERROR! FAILED TO AUTO-DECREMENT R4
002250  022705  120002                  CMP     #UI5+2,R5       ;CHECK THAT R5 WAS UNCHANGED
002254  001401                          BEQ     .+4
002256  000000                          HLT                     ;ERROR!
002260  104000                          SCOPE                   ;SCOPE STORES PC IN R1 & SETS ALL STACK PTRS

                                ;CHECK ABORT AT SRC08
                                ;ABORTS WHEN SOURCE OPERAND IS FETCHED
                                ;SOURCE MODE = 6
002262  012737  002306  000250          MOV     #T6C,@#MMVEC    ;LOAD MEM MGMT ERROR VECTOR
002270  012702  177777                  MOV     #-1,R2
002274  005237  177572                  INC     @#SR0           ;ENABLE MEMORY MGMT
002300  016702  014374          T6A!    MOV     KI0,R2          ;SEG LENGTH ABORT AT SRC08
002304  000000                  T6B!    HLT                     ;ERROR! FAILED TO ABORT
002306                          T6C!
002306  022767  040001  175256          CMP     #PLA+1,SR0      ;CHECK SR0 (ABORT CONDITIONS
002314  001401                          BEQ     .+4             ;& FAILING PAGE #)
```

```
002316  000000                          HLT                     ;ERROR! INCORRECT ABORT CONDITIONS OR PAGE IDENT
002320  022767  002300  175250          CMP     #T6A,SR2        ;CHECK CONTENTS OF SR2
002326  001401                          BEQ     .+4             ;(PC OF ABORTED INSTRUCTION)
002330  000000                          HLT                     ;ERROR! INCORRECT PC IN SR2
002332  005202                          INC     R2              ;CHECK THAT R2 WAS NOT CHANGED
002334  001401                          BEQ     .+4
002336  000000                          HLT                     ;ERROR!
002340  104000                          SCOPE                   ;SCOPE STORES PC IN R1 & SETS ALL STACK PTRS

                                ;CHECK ABORT AT SRC13
                                ;ABORTS WHEN ADDRESS OF SOURCE OPERAND IS FETCHED
                                ;SOURCE MODE = 7, PC
002342  012737  002364  000250          MOV     #T7C,@#MMVEC    ;LOAD MEM MGMT ERROR VECTOR
002350  005004                          CLR     R4
002352  005237  177572                  INC     @#SR0           ;ENABLE MEMORY MGMT
002356  067404  016700          T7A:    ADD     @K10(R4),R4     ;SEG LEN ABORT AT SRC13
002362  000000          T7B:    HLT                             ;ERROR! FAILED TO ABORT
002364                  T7C:
002364  022767  040001  175200          CMP     #PLA+1,SR0      ;CHECK SR0 (ABORT CONDITIONS
002372  001401                          BEQ     .+4             ;& FAILING PAGE #)
002374  000000                          HLT                     ;ERROR! INCORRECT ABORT CONDITIONS OR PAGE IDENT
002376  022767  002356  175172          CMP     #T7A,SR2        ;CHECK CONTENTS OF SR2
002404  001401                          BEQ     .+4             ;(PC OF ABORTED INSTRUCTION)
002406  000000                          HLT                     ;ERROR! INCORRECT PC IN SR2
002410  005704                          TST     R4
002412  001401                          BEQ     .+4
002414  000000                          HLT
002416  104000                          SCOPE                   ;SCOPE STORES PC IN R1 & SETS ALL STACK PTRS

                                ;CHECK ABORT AT SRC15
                                ;ABORTS WHEN SOURCE OPERAND IS FETCHED
                                ;SOURCE MODE = 3,PC
002420  012737  002442  000250          MOV     #T10C,@#MMVEC   ;LOAD MEM MGMT ERROR VECTOR
002426  005003                          CLR     R3
002430  005237  177572                  INC     @#SR0           ;ENABLE MEMORY MGMT
002434  013703  016700          T10A:   MOV     @#K10,R3        ;SEG LEN ABORT AT SRC15
002440  000000          T10B:   HLT                             ;ERROR! FAILED TO ABORT
002442                  T10C:
002442  022767  040001  175122          CMP     #PLA+1,SR0      ;CHECK SR0 (ABORT CONDITIONS
002450  001401                          BEQ     .+4             ;& FAILING PAGE #)
002452  000000                          HLT                     ;ERROR! INCORRECT ABORT CONDITIONS OR PAGE IDENT
002454  022767  002434  175114          CMP     #T10A,SR2       ;CHECK CONTENTS OF SR2
002462  001401                          BEQ     .+4             ;(PC OF ABORTED INSTRUCTION)
002464  000000                          HLT                     ;ERROR! INCORRECT PC IN SR2
002466  005703                          TST     R3
002470  001401                          BEQ     .+4
002472  000000                          HLT                     ;ERROR!
002474  104000                          SCOPE                   ;SCOPE STORES PC IN R1 & SETS ALL STACK PTRS


                                ;CHECK ABORT AT RTI,00
                                ;ABORTS WHEN TOP WORD OFF STACK (PC) IS FETCHED
002476  012737  002544  000250          MOV     #T13C,@#MMVEC   ;LOAD MEM MGMT ERROR VECTOR
002504  012767  170000  175264          MOV     #UM+PUM,PSW     ;USER MODE!!! PREV USER MODE!!
```

```
002512  012706  040100                  MOV     #UI2+100,USP    ;USER STACK PTR IS NON-RES
002516  012737  002542  017100          MOV     #T13D,@#PUI2+100        ;LOAD 'NEW' PC
002524  005037  017102                  CLR     @#PUI2+102
002530  005237  177572                  INC     @#SR0           ;ENABLE MEMORY MGMT
002534  000277                          SCC
002536  000002                  T13A:   RTI                     ;NON-RES ABORT AT RTI,00
002540  000000                  T13B:   HLT                     ;ERROR: FAILED TO ABORT
002542  000000                  T13D:   HLT                     ;ERROR: RTI FAILED & DID NOT ABORT
002544  022706  001074          T13C:   CMP     #KPTR+4,KSP     ;CHECK STACK PTR
002550  001401                          BEQ     .+4
002552  000000                          HLT
002554  022766  170017  000002          CMP     #UM+PUM+17,2(KSP)       ;CHECK THAT CORRECT STATUS
002562  001401                          BEQ     .+4             ;WAS SAVED ON THE STACK
002564  000000                          HLT                     ;ERROR: INCORRECT STATUS
002566  022767  040145  174776          CMP     #PLA+UPG+VS2+1,SR0      ;CHECK SR0 (ABORT CONDITIONS
002574  001401                          BEQ     .+4             ;& FAILING PAGE #)
002576  000000                          HLT                     ;ERROR: INCORRECT ABORT CONDITIONS OR PAGE IDENT
002600  022767  002536  174770          CMP     #T13A,SR2       ;CHECK CONTENTS OF SR2
002606  001401                          BEQ     .+4             ;(PC OF ABORTED INSTRUCTION)
002610  000000                          HLT                     ;ERROR: INCORRECT PC IN SR2
002612  106506                          MFPD    USP             ;PUSH USER STACK PTR ONTO KERNEL STACK
002614  022716  040102                  CMP     #UI2+102,(KSP)  ;CHECK THAT USER STACK PTR WAS POPPED
002620  001401                          BEQ     .+4
002622  000000                          HLT                     ;ERROR:
002624  104000                          SCOPE                   ;SCOPE STORES PC IN R1 & SETS ALL STACK PTRS

                                ;CHECK ABORT AT RTI,01
                                ;ABORTS WHEN SECOND WORD ON STACK (STATUS) IS FETCHED
002626  012737  002672  000250          MOV     #T14C,@#MMVEC   ;LOAD MEM MGMT ERROR VECTOR
002634  012767  170000  175134          MOV     #UM+PUM,PSW     ;USER MODE!!!,PREV USER MODE!!
002642  012706  100076                  MOV     #UI4+76,USP
002646  012737  002670  017376          MOV     #T14D,@#PUI4+76 ;LOAD USER STACK (PHYS ADRS,)
002654  005037  017400                  CLR     @#PUI4+100      ;AND 'NEW' STATUS
002660  005237  177572                  INC     @#SR0           ;ENABLE MEMORY MGMT
002664  000006                  T14A:   RTT                     ;SEG LEN ABORT AFTER FIRST POP RTI,01
002666  000000                  T14B:   HLT                     ;ERROR: FAILED TO ABORT
002670  000000                  T14D:   HLT                     ;ERROR:
002672                          T14C:
002672  022767  040151  174672          CMP     #PLA+UPG+VS4+1,SR0      ;CHECK SR0 (ABORT CONDITIONS
002700  001401                          BEQ     .+4             ;& FAILING PAGE #)
002702  000000                          HLT                     ;ERROR: INCORRECT ABORT CONDITIONS OR PAGE IDENT
002704  022767  002664  174664          CMP     #T14A,SR2       ;CHECK CONTENTS OF SR2
002712  001401                          BEQ     .+4             ;(PC OF ABORTED INSTRUCTION)
002714  000000                          HLT                     ;ERROR: INCORRECT PC IN SR2
002716  106506                          MFPD    USP             ;PUSH USER STACK PTR ONTO KERNEL STACK
002720  022716  100102                  CMP     #UI4+102,(KSP)  ;CHECK THAT USER STACK PTR POPPED TWICE
002724  001401                          BEQ     .+4
002726  000000                          HLT
002730  104000                          SCOPE                   ;SCOPE STORES PC IN R1 & SETS ALL STACK PTRS

                                ;CHECK ABORT AT RTS,02
                                ;ABORTS WHEN TOP WORD ON USER STACK (RETURN PC) IS FETCHED
002732  012767  170000  175036          MOV     #UM+PUM,PSW     ;USER MODE!!! PREV USER MODE!!
002740  012706  020100                  MOV     #UI1+100,USP
```

```
002744  012737  002770  000250          MOV     #T16C,@#MMVEC   ;LOAD MEM MGMT ERROR VECTOR
002752  012705  002766                  MOV     #T16D,R5
002756  005237  177572                  INC     @#SR0           ;ENABLE MEMORY MGMT
002762  000205                  T16A:   RTS     5               ;ABORTS AT RTS;02 (STACK IS NON-RES)
002764  000000                  T16B:   HLT                     ;ERROR: RTS& ABORT FAILED
002766  000000                  T16D:   HLT                     ;ERROR: ABORT FAILED
002770  022706  001074          T16C:   CMP     #KPTR-4,KSP     ;CHECK STACK PTR
002774  001401                          BEQ     .+4
002776  000000                          HLT
003000  022767  040143  174564          CMP     #PLA+UPG+VS1+1,SR0      ;CHECK SR0 (ABORT CONDITIONS
003006  001401                          BEQ     .+4             ;& FAILING PAGE #)
003010  000000                          HLT                     ;ERROR: INCORRECT ABORT CONDITIONS OR PAGE IDENT
003012  022767  002762  174556          CMP     #T16A,SR2       ;CHECK CONTENTS OF SR2
003020  001401                          BEQ     .+4             ;(PC OF ABORTED INSTRUCTION)
003022  000000                          HLT                     ;ERROR: INCORRECT PC IN SR2
003024  022705  002766                  CMP     #T16D,R5        ;CHECK THAT R5 DID NOT CHANGE
003030  001401                          BEQ     .+4
003032  000000                          HLT                     ;ERROR!
003034  106506                          MFPD    USP             ;PUSH USER STACK PTR ONTO KERNEL STACK
003036  022716  020102                  CMP     #UI1+102,(KSP)  ;CHECK THAT USER STACK WAS POPPED
003042  001401                          BEQ     .+4
003044  000000                          HLT                     ;ERROR: INCORRECT USER STACK PTR
003046  104000                          SCOPE                   ;SCOPE STORES PC IN R1 & SETS ALL STACK PTRS
                                ;CHECK ABORT AT SRC07
                                ;ABORTS WHEN SOURCE INDEX IS FETCHED
                                ;SOURCE MODE = 6, PC
003050  012737  003112  000250          MOV     #T20C,@#MMVEC   ;LOAD MEM MGMT ERROR VECTOR
003056  012702  177777                  MOV     #-1,R2          ;PRESET DEST REG
003062  012737  016702  016676          MOV     #16702,@#KI0-2  ;16702,000000 IS A MOV .+4,R2
003070  005037  016700                  CLR     @#KI0           ;INSTRUCTION
003074  005037  016702                  CLR     @#KI0+2
003100  005237  177572                  INC     @#SR0           ;ENABLE MEMORY MGMT
003104  000277                          SCC                     ;PRESET CC'S
003106  000137  016676                  JMP     @#KI0-2         ;GO TO MOV INST,
        003112                          RETURN=.
                                ;************************ NOTE PC CHANGE **********************************
        016676                          .=KI0-2
016676  016702  000000          T20A:   MOV     .+4,R2          ;SEG LEN ABORT WHEN INDEX VALUE IS FETCHED
016702  000000                  T20B:   HLT                             ;ERROR: FAILED TO ABORT
                                ;************************ RETURN PC ***************************************
        003112                          .=RETURN
003112  022706  001074          T20C:   CMP     #KPTR-4,KSP     ;CHECK STACK PTR
003116  001401                          BEQ     .+4
003120  000000                          HLT
003122  022766  000017  000002          CMP     #17,2(KSP)      ;CHECK THAT CORRECT STATUS
003130  001401                          BEQ     .+4             ;WAS SAVED ON THE STACK
003132  000000                          HLT                     ;ERROR: INCORRECT STATUS
003134  022767  040001  174430          CMP     #PLA+IS+VS0+1,SR0       ;CHECK SR0 (ABORT CONDITIONS
003142  001401                          BEQ     .+4             ;& FAILING PAGE #)
003144  000000                          HLT                     ;ERROR: INCORRECT ABORT CONDITIONS OR PAGE IDENT
003146  022767  016676  174422          CMP     #T20A,SR2       ;CHECK CONTENTS OF SR2
003154  001401                          BEQ     .+4             ;(PC OF ABORTED INSTRUCTION)
003156  000000                          HLT                     ;ERROR: INCORRECT PC IN SR2
003160  005202                          INC     R2
```

```
003162  001401                          BEQ     .+4
003164  000000                          HLT
003166  104000                          SCOPE                   ;SCOPE STORES PC IN R1 & SETS ALL STACK PTRS

                                ;CHECK ABORT AT SRC10
                                ;ABORTS WHEN SOURCE INDEX IS FETCHED
                                ;SOURCE MODE = 7
003170  012737  003232  000250          MOV     #T21C,@#MMVEC   ;LOAD MEM MGMT ERROR VECTOR
003176  012737  177777  016700          MOV     #-1,@#PKI6
003204  012702  140000                  MOV     #KI6,R2         ;LOAD INDEX REGISTER
003210  012737  017202  016676          MOV     #017202,@#KI0-2 ;017202,000000 IS A MOV  @0(R2),R2
003216  005037  016700                  CLR     @#KI0           ;INSTRUCTION
003222  005237  177572                  INC     @#SR0           ;ENABLE MEMORY MGMT
003226  000137  016676                  JMP     @#KI0-2
        003232                          RETURN#,
                                ;************************** NOTE PC CHANGE **************************
        016676                          ,#KI0-2
016676  017202  000000          T21A:   MOV     @0(R2),R2       ;SEG LEN ABORT AT SRC10
016702  000000          T21B:   HLT                             ;ERROR: FAILED TO ABORT
        003232                          ,#RETURN
                                ;************************ RETURN PC *********************************

003232                          T21C:
003232  022767  040001  174332          CMP     #PLA+IS+VS0+1,SR0       ;CHECK SR0 (ABORT CONDITIONS
003240  001401                          BEQ     .+4                     ;& FAILING PAGE #)
003242  000000                          HLT                             ;ERROR: INCORRECT ABORT CONDITIONS OR PAGE IDENT
003244  022767  016676  174324          CMP     #T21A,SR2               ;CHECK CONTENTS OF SR2
003252  001401                          BEQ     .+4                     ;(PC OF ABORTED INSTRUCTION)
003254  000000                          HLT                             ;ERROR: INCORRECT PC IN SR2
003256  022702  140000                  CMP     #KI6,R2 ;CHECK THAT R2 IS UNCHANGED
003262  001401                          BEQ     .+4
003264  000000                          HLT                             ;ERROR!
003266  104000                          SCOPE                   ;SCOPE STORES PC IN R1 & SETS ALL STACK PTRS

                                ;CHECK ABORT AT FET03
                                ;ABORTS WHEN INST FOLLOWING SOB IS FETCHED
003270  012703  000001                  MOV     #1,R3
003274  012737  077302  016676          MOV     #077302,@#KI0-2 ;077302=SOB  R3,.-2
003302  005037  016674                  CLR     @#KI0-4         ;CLEAR INST, PRECEDING SOB (.-2)
003306  005037  016700                  CLR     @#KI0           ;PUT HLT FOLLOWING SOB
003312  012737  003332  000250          MOV     #T22C,@#MMVEC   ;LOAD MEM MGMT ERROR VECTOR
003320  005237  177572                  INC     @#SR0           ;ENABLE MEMORY MGMT
003324  000277                          SCC                     ;PRESET CC'S
003326  000137  016676                  JMP     @#KI0-2         ;GO TO SOB INST.

        003332                          RETURN#,
        016674                          ,#KI0-4
016674  000000          T22:    HLT                             ;ERROR: SOB BRANCHED & FAILED TO ABORT
016676  077302          T22A:   SOB     R3,.-2                  ;ABORTS WHEN NEXT INST, IS FETCHED
016700  000000          T22AA:  HLT                             ;ERROR: FAILED TO ABORT
016702  000000          T22B:   0
        003332                          ,#RETURN

003332  022706  001074          T22C:   CMP     #KPTR-4,KSP     ;CHECK STACK PTR
```

```
003336  001401                          BEQ     .+4
003340  000000                          HLT
003342  022766  000017  000002          CMP     #17,2(KSP)      ;CHECK THAT CORRECT STATUS
003350  001401                          BEQ     .+4             ;WAS SAVED ON THE STACK
003352  000000                          HLT                     ;ERROR: INCORRECT STATUS
003354  022767  040001  174210          CMP     #PLA+VS0+1,SR0  ;CHECK SR0 (ABORT CONDITIONS
003362  001401                          BEQ     .+4             ;& FAILING PAGE #)
003364  000000                          HLT                     ;ERROR: INCORRECT ABORT CONDITIONS OR PAGE IDENT
003366  022767  016676  174202          CMP     #T22A,SR2       ;CHECK CONTENTS OF SR2
003374  001401                          BEQ     .+4             ;(PC OF ABORTED INSTRUCTION)
003376  000000                          HLT                     ;ERROR: INCORRECT PC IN SR2
003400  005703                          TST     R3              ;CHECK THAT R3 DECREMENTD
003402  001401                          BEQ     .+4
003404  000000                          HLT                     ;ERROR: R3 WAS NOT DECREMENTED BY SOB
003406  104000                          SCOPE                   ;SCOPE STORES PC IN R1 & SETS ALL STACK PTRS

                                ;CHECK ABORT AT MOV16
                                ;ABORTS WHEN DEST OPERAND IS FETCHED
003410  012767  030000  174360          MOV     #KM+PUM,PSW
003416  012737  003440  000250          MOV     #T24C,@#MMVEC   ;LOAD MEM MGMT ERROR VECTOR
003424  012702  040000                  MOV     #UI2,R2
003430  005237  177572                  INC     @#SR0           ;ENABLE MEMORY MGMT
003434  106542                  T24A:   MFPD    -(R2)           ;NON-RESIDENT ABORT AT MOV16
003436  000000                  T24B:   HLT
003440  022706  001074          T24C:   CMP     #KPTR+4,KSP     ;CHECK STACK PTR
003444  001401                          BEQ     .+4
003446  000000                          HLT
003450  022767  040143  174114          CMP     #UPG+PLA+VS1+1,SR0      ;CHECK SR0 (ABORT CONDITIONS
003456  001401                          BEQ     .+4             ;& FAILING PAGE #)
003460  000000                          HLT                     ;ERROR: INCORRECT ABORT CONDITIONS OR PAGE IDENT
003462  022767  003434  174106          CMP     #T24A,SR2       ;CHECK CONTENTS OF SR2
003470  001401                          BEQ     .+4             ;(PC OF ABORTED INSTRUCTION)
003472  000000                          HLT                     ;ERROR: INCORRECT PC IN SR2
003474  022702  037776                  CMP     #UI2-2,R2       ;CHECK THAT R2 AUTO-DECREMENTED
003500  001401                          BEQ     .+4
003502  000000                          HLT                     ;ERROR: R2 DID NOT AUTO-DECREMENT
003504  104000                          SCOPE                   ;SCOPE STORES PC IN R1 & SETS ALL STACK PTRS

                                ;CHECK ABORT AT DST00
                                ;ABORTS WHEN DEST OPERAND IS FETCHED
003506  012737  003536  000250          MOV     #T25C,@#MMVEC   ;LOAD MEM MGMT ERROR VECTOR
003514  012702  177572                  MOV     #SR0,R2
003520  012767  170000  174250          MOV     #UM+PUM,PSW
003526  005237  177572                  INC     @#SR0           ;ENABLE MEMORY MGMT
003532  005012                  T25A:   CLR     (R2)            ;ABORT AT DST00
003534  000000                  T25B:   HLT                     ;ERROR: FAILED TO ABORT
003536  022706  001074          T25C:   CMP     #KPTR+4,KSP     ;CHECK STACK PTR
003542  001401                          BEQ     .+4
003544  000000                          HLT
003546  022767  140157  174016          CMP     #NRA+PLA+UPG+VS7+1,SR0  ;CHECK SR0 (ABORT CONDITIONS
003554  001401                          BEQ     .+4             ;& FAILING PAGE #)
003556  000000                          HLT                     ;ERROR: INCORRECT ABORT CONDITIONS OR PAGE IDENT
003560  022767  003532  174010          CMP     #T25A,SR2       ;CHECK CONTENTS OF SR2
003566  001401                          BEQ     .+4             ;(PC OF ABORTED INSTRUCTION)
```

```
003570  000000                          HLT                     ;ERROR: INCORRECT PC IN SR2
003572  104000                          SCOPE                   ;SCOPE STORES PC IN R1 & SETS ALL STACK PTRS

                                ;CHECK ABORT AT DST01
                                ;ABORTS WHEN DEST OPERAND IS FETCHED
003574  012737  003624  000250          MOV     #T30C,@#MMVEC   ;LOAD MEM MGMT ERROR VECTOR
003602  012703  016677                  MOV     #K10-1,R3
003606  012737  177777  016700          MOV     #-1,@#K10
003614  005237  177572                  INC     @#SR0           ;ENABLE MEMORY MGMT
003620  142323                  T30A:   BICB    (R3)+,(R3)+     ;SEG LENGTH ABORT AT DST01
003622  000000                          HLT                     ;ERROR: FAILED TO ABORT

003624                          T30C:
003624  022767  040001  173740          CMP     #PLA+1,SR0      ;CHECK SR0 (ABORT CONDITIONS
003632  001401                          BEQ     .+4             ;& FAILING PAGE #)
003634  000000                          HLT                     ;ERROR: INCORRECT ABORT CONDITIONS OR PAGE IDENT
003636  022767  003620  173732          CMP     #T30A,SR2       ;CHECK CONTENTS OF SR2
003644  001401                          BEQ     .+4             ;(PC OF ABORTED INSTRUCTION)
003646  000000                          HLT                     ;ERROR: INCORRECT PC IN SR2
003650  005037  177572                  CLR     @#SR0           ;DISABLE MEMORY MGMT
003654  022703  016701                  CMP     #K10+1,R3       ;CHECK AUTO-INC TWICE
003660  001401                          BEQ     .+4
003662  000000                          HLT                     ;ERROR:
003664  005237  016700                  INC     @#K10
003670  001401                          BEQ     .+4
003672  000000                          HLT                     ;ERROR:
003674  104000                          SCOPE                   ;SCOPE STORES PC IN R1 & SETS ALL STACK PTRS

                                ;CHECK ABORT AT DST04
                                ;ABORTS WHEN ADDRESS OF DEST OPERAND IS FETCHED
003676  012737  003746  000250          MOV     #T31C,@#MMVEC   ;LOAD MEM MGMT ERROR VECTOR
003704  012702  040000                  MOV     #UI2,R2
003710  012703  017100                  MOV     #PUI1,R3
003714  012713  177777                  MOV     #-1,(R3)
003720  011337  017000                  MOV     (R3),@#PUI2
003724  012703  020002                  MOV     #UI1+2,R3       ;R3= USER VIRTUAL ADDRESS
003730  012767  170000  174040          MOV     #UM+PUM,PSW
003736  005237  177572                  INC     @#SR0           ;ENABLE MEMORY MGMT
003742  114332                  T31A:   MOVB    -(R3),@(R2)+    ;NON-RESIDENT ABORT AT DST04
003744  000000                  T31B:   HLT                     ;ERROR: FAILED TO ABORT

003746                          T31C:
003746  022767  140157  173616          CMP     #NRA+PLA+UPG+VS7+1,SR0  ;CHECK SR0 (ABORT CONDITIONS
003754  001401                          BEQ     .+4             ;& FAILING PAGE #)
003756  000000                          HLT                     ;ERROR: INCORRECT ABORT CONDITIONS OR PAGE IDENT
003760  022767  003742  173610          CMP     #T31A,SR2       ;CHECK CONTENTS OF SR2
003766  001401                          BEQ     .+4             ;(PC OF ABORTED INSTRUCTION)
003770  000000                          HLT                     ;ERROR: INCORRECT PC IN SR2
003772  022702  040002                  CMP     #UI2+2,R2       ;CHECK AUTO-INC
003776  001401                          BEQ     .+4
004000  000000                          HLT                     ;ERROR:
004002  022703  020001                  CMP     #UI1+1,R3       ;CHECK AUTO DECREMENT OF R3
004006  001401                          BEQ     .+4
004010  000000                          HLT                     ;ERROR: R3 NOT AUTO-DECREMENTED
```

```
004012  104000                          SCOPE                   ;SCOPE STORES PC IN R1 & SETS ALL STACK PTRS

                                ;CHECK ABORT AT MOV03
004014  012737  004060  000250          MOV     #T32C,@#MMVEC   ;LOAD MEM MGMT ERROR VECTOR
004022  012767  170000  173746          MOV     #UM+PUM,PSW
004030  012706  000600                  MOV     #UPTR,USP
004034  005016                          CLR     (USP)
004036  012702  060000                  MOV     #UI3,R2
004042  012737  177777  017400          MOV     #-1,@#PUI3
004050  005237  177572                  INC     @#SR0           ;ENABLE MEMORY MGMT
004054  006632                  T32A:   MTPI    @(R2)+          ;NON-RESIDENT ABORT AT MOV03
004056  000000                  T32B:   HLT                     ;ERROR: FAILED TO ABORT
004060                          T32C:
004060  022767  100147  173504          CMP     #NRA+UPG+VS3+1,SR0      ;CHECK SR0 (ABORT CONDITIONS
004066  001401                          BEQ     .+4             ;& FAILING PAGE #)
004070  000000                          HLT                     ;ERROR: INCORRECT ABORT CONDITIONS OR PAGE IDENT
004072  022767  004054  173476          CMP     #T32A,SR2       ;CHECK CONTENTS OF SR2
004100  001401                          BEQ     .+4             ;(PC OF ABORTED INSTRUCTION)
004102  000000                          HLT                     ;ERROR: INCORRECT PC IN SR2
004104  106506                          MFPD    USP             ;PUSH USER STACK PTR ONTO KERNEL STACK
004106  022716  000602                  CMP     #UPTR+2,(KSP)   ;CHECK THAT USER STACK PTR POPPED
004112  001401                          BEQ     .+4
004114  000000                          HLT                     ;ERROR:
004116  022702  060002                  CMP     #UI3+2,R2       ;CHECK AUTO-INC
004122  001401                          BEQ     .+4
004124  000000                          HLT                     ;ERROR:
004126  104000                          SCOPE                   ;SCOPE STORES PC IN R1 & SETS ALL STACK PTRS

                                ;CHECK ABORT AT MOV11
                                ;ABORTS WHEN ADDRESS OF DEST OPERAND IS FETCHED
004130  012737  004172  000250          MOV     #T33C,@#MMVEC   ;LOAD MEM MGMT ERROR VECTOR
004136  012767  030000  173632          MOV     #KM+PUM,PSW
004144  012716  117776                  MOV     #UI5-2,(KSP)    ;NON-RES ADDRESS
004150  012746  140000                  MOV     #KI6,-(KSP)     ;ADDRESS POINTER
004154  012737  117776  140000          MOV     #UI5-2,@#KI6    ;NON-RESIDENT ADDRESS
004162  005237  177572                  INC     @#SR0           ;ENABLE MEMORY MGMT
004166  106636                  T33A:   MTPD    @(KSP)+         ;NON-RESIDENT ABORT AT MOV11 WHEN MTPD
                                                                ;ADDRESSES FINAL ADDRESS
004170  000000                  T33B:   HLT                     ;ERROR: FAILED TO ABORT
004172                          T33C:
004172  022767  040151  173372          CMP     #PLA+UPG+VS4+1,SR0      ;CHECK SR0 (ABORT CONDITIONS
004200  001401                          BEQ     .+4             ;& FAILING PAGE #)
004202  000000                          HLT                     ;ERROR: INCORRECT ABORT CONDITIONS OR PAGE IDENT
004204  022767  004166  173364          CMP     #T33A,SR2       ;CHECK CONTENTS OF SR2
004212  001401                          BEQ     .+4             ;(PC OF ABORTED INSTRUCTION)
004214  000000                          HLT                     ;ERROR: INCORRECT PC IN SR2
004216  106506                          MFPD    KSP             ;GET KERNEL STACK PTR
004220  022716  001076                  CMP     #KPTR+2,(KSP)   ;CHECK THAT KERNEL STACK PTR POPPED TWICE
004224  001401                          BEQ     .+4
004226  000000                          HLT                     ;ERROR:
004230  104000                          SCOPE                   ;SCOPE STORES PC IN R1 & SETS ALL STACK PTRS

                                ;CHECK ABORT AT MOV15
                                ;ABORTS WHEN ADDRESS OF DEST OPERAND IS FETCHED
```

```
                                        ;DM=5
004232  012737  004266  000250            MOV     #T35C,@#MMVEC   ;LOAD MEM MGMT ERROR VECTOR
004240  012704  016702                    MOV     #KI0+2,R4
004244  012727  020000  016676            MOV     #20000,#KI0-2
004252  005237  177572                    INC     @#SR0           ;ENABLE MEMORY MGMT
004256  000277                            SCC
004260  112754  177777          T35A:     MOVB    #-1,@-(R4)      ;SEG LENGTH ABORT AT MOV15
004264  000000                  T35B:     HLT                     ;ERROR: FAILED TO ABORT
004266                          T35C:
004266  022766  000017  000002            CMP     #17,2(KSP)      ;CHECK THAT CORRECT STATUS
004274  001401                            BEQ     .+4             ;WAS SAVED ON THE STACK
004276  000000                            HLT                     ;ERROR: INCORRECT STATUS
004300  022767  040001  173264            CMP     #PLA+VS0+1,SR0  ;CHECK SR0 (ABORT CONDITIONS
004306  001401                            BEQ     .+4             ;& FAILING PAGE #)
004310  000000                            HLT                     ;ERROR: INCORRECT ABORT CONDITIONS OR PAGE IDENT
004312  022767  004260  173256            CMP     #T35A,SR2       ;CHECK CONTENTS OF SR2
004320  001401                            BEQ     .+4             ;(PC OF ABORTED INSTRUCTION)
004322  000000                            HLT                     ;ERROR: INCORRECT PC IN SR2
004324  022704  016700                    CMP     #KI0,R4
004330  001401                            BEQ     .+4
004332  000000                            HLT                     ;ERROR:
004334  104000                            SCOPE                   ;SCOPE STORES PC IN R1 & SETS ALL STACK PTRS

                                        ;CHECK ABORT AT DST15
                                        ;ABORTS WHEN DEST OPERAND IS FETCHED
                                        ;DM=4
004336  012737  004370  000250            MOV     #T36C,@#MMVEC   ;LOAD MEM MGMT ERROR VECTOR
004344  012704  140002                    MOV     #KI6+2,R4
004350  012703  016702                    MOV     #KI0+2,R3
004354  012713  177777                    MOV     #-1,(R3)
004360  005237  177572                    INC     @#SR0           ;ENABLE MEMORY MGMT
004364  154443                  T36A:     BISB    -(R4),-(R3)     ;SEG LENGTH ABORT AT DST15
004366  000000                  T36B:     HLT                     ;ERROR: FAILED TO ABORT

004370                          T36C:
004370  022767  040001  173174            CMP     #PLA+1,SR0      ;CHECK SR0 (ABORT CONDITIONS
004376  001401                            BEQ     .+4             ;& FAILING PAGE #)
004400  000000                            HLT                     ;ERROR: INCORRECT ABORT CONDITIONS OR PAGE IDENT
004402  022767  004364  173166            CMP     #T36A,SR2       ;CHECK CONTENTS OF SR2
004410  001401                            BEQ     .+4             ;(PC OF ABORTED INSTRUCTION)
004412  000000                            HLT                     ;ERROR: INCORRECT PC IN SR2
004414  022703  016701                    CMP     #KI0+1,R3       ;CHECK AUTO-DEC
004420  001401                            BEQ     .+4
004422  000000                            HLT                     ;ERROR:
004424  022704  140001                    CMP     #KI6+1,R4       ;CHECK AUTO-DEC
004430  001401                            BEQ     .+4
004432  000000                            HLT                     ;ERROR: AUTO-DEC FAILED
004434  104000                            SCOPE                   ;SCOPE STORES PC IN R1 & SETS ALL STACK PTRS


                                        ;CHECK ABORT AT MOV08
                                        ;WHEN INSTRUCTION FETCHES DESTINATION INDEX VALUE
004436  012737  004504  000250            MOV     #T40C,@#MMVEC   ;LOAD MEM MGMT ERROR VECTOR
004444  012767  170000  173324            MOV     #UM+PUM,PSW
```

```
004452  012737  113767  017074          MOV     #113767,@#PUI2+74       ;113767,020001,177776
004460  012737  020001  017076          MOV     #20001,@#PUI2+76        ;IS A MOVB  @#20001,.+4
004466  012737  177776  017100          MOV     #177776,@#PUI2+100      ;INSTRUCTION
004474  005237  177572                  INC     @#SR0           ;ENABLE MEMORY MGMT
004500  000137  040074                  JMP     @#UI2+74
        004504                          RETURN=.
        017074                          .=PUI2+74
017074  113767  020001  177776  T40A:   MOVB    @#20001,.+4     ;SEG LENGTH ABORT WHEN INST. FETCHES
                                                                ;DEST INDEX WORD AT MOV08
017102  000000                  T40B:   HLT                     ;ERROR: FAILED TO ABORT
        004504                          .=RETURN

004504                          T40C:
004504  022767  040145  173060          CMP     #PLA+UPG+VS2+1,SR0      ;CHECK SR0 (ABORT CONDITIONS
004512  001401                          BEQ     .+4             ;& FAILING PAGE #)
004514  000000                          HLT                     ;ERROR: INCORRECT ABORT CONDITIONS OR PAGE IDENT
004516  022767  040074  173052          CMP     #UI2+74,SR2     ;CHECK CONTENTS OF SR2
004524  001401                          BEQ     .+4             ;(PC OF ABORTED INSTRUCTION)
004526  000000                          HLT                     ;ERROR: INCORRECT PC IN SR2
004530  104000                          SCOPE                   ;SCOPE STORES PC IN R1 & SETS ALL STACK PTRS

                                ;CHECK ABORT AT MOV17
                                ;WHEN INSTRUCTION FETCHES DESTINATION OPERAND
004532  012737  004600  000250          MOV     #T41C,@#MMVEC   ;LOAD MEM MGMT ERROR VECTOR
004540  012767  170000  173230          MOV     #UM+PUM,PSW     ;USER MODE!!!,PREV USER MODE!!
004546  012703  100000                  MOV     #UI4,R3
004552  012704  100102                  MOV     #UI4+102,R4
004556  012737  010344  017200          MOV     #010344,@#PUI5  ;012344 = MOV  R3,-(R4)
004564  005037  017202                  CLR     @#PUI5+2
004570  005237  177572                  INC     @#SR0           ;ENABLE MEMORY MGMT
004574  000137  120000                  JMP     @#UI5
        004600                          RETURN=.
        017200                          .=PUI5
017200  010344                  T41A:   MOV     R3,-(R4)        ;ABORT AT MOV17
017202  000000                  T41B:   HLT             ;ERROR: FAILED TO ABORT
        004600                          .=RETURN

004600                          T41C:
004600  022767  040151  172764          CMP     #PLA+UPG+VS4+1,SR0      ;CHECK SR0 (ABORT CONDITIONS
004606  001401                          BEQ     .+4             ;& FAILING PAGE #)
004610  000000                          HLT                     ;ERROR: INCORRECT ABORT CONDITIONS OR PAGE IDENT
004612  022767  120000  172756          CMP     #UI5,SR2        ;CHECK CONTENTS OF SR2
004620  001401                          BEQ     .+4             ;(PC OF ABORTED INSTRUCTION)
004622  000000                          HLT                     ;ERROR: INCORRECT PC IN SR2
004624  022704  100100                  CMP     #UI4+100,R4
004630  001401                          BEQ     .+4
004632  000000                          HLT
004634  104000                          SCOPE                   ;SCOPE STORES PC IN R1 & SETS ALL STACK PTRS

                                ;CHECK ABORT AT MOV16
                                ;(WHEN INSTRUCTION FETCHES ADDRESS OF DESTINATION OPERAND)
004636  012737  004674  000250          MOV     #T42C,@#MMVEC   ;LOAD MEM MGMT ERROR VECTOR
004644  012767  030000  173124          MOV     #KM+PUM,PSW
004652  012703  140102                  MOV     #KI6+102,R3
```

```
004656  012737  177777  017000          MOV     #-1,@#PKI6+100
004664  005237  177572                  INC     @#SR0           ;ENABLE MEMORY MGMT
004670  106653                  T42A:   MTPD    @-(R3)          ;SEG LENGTH ABORT AT MOV16
004672  000000                  T42B:   HLT                     ;ERROR: FAILED TO ABORT

004674  022706  001076          T42C:   CMP     #KPTR-2,KSP     ;CHECK STACK PTR ( 1 POP, 2 PUSHES)
004700  001401                          BEQ     .+4
004702  000000                          HLT                     ;ERROR: INCORRECT STACK PTR
004704  022767  040015  172660          CMP     #PLA+VS6+1,SR0  ;CHECK SR0 (ABORT CONDITIONS
004712  001401                          BEQ     .+4             ;& FAILING PAGE #)
004714  000000                          HLT                     ;ERROR: INCORRECT ABORT CONDITIONS OR PAGE IDENT
004716  022767  004670  172652          CMP     #T42A,SR2       ;CHECK CONTENTS OF SR2
004724  001401                          BEQ     .+4             ;(PC OF ABORTED INSTRUCTION)
004726  000000                          HLT                     ;ERROR: INCORRECT PC IN SR2
004730  022703  140100                  CMP     #KI6+100,R3     ;CHECK AUTO-DECREMENT
004734  001401                          BEQ     .+4
004736  000000                          HLT                     ;ERROR: DID NOT AUTO-DEC
004740  104000                          SCOPE                   ;SCOPE STORES PC IN R1 & SETS ALL STACK PTRS

                                ;CHECK ABORT AT DST04
                                ;ABORTS WHEN ADDRESS TO JUMP TO IS FETCHED
004742  012737  005010  000250          MOV     #T43C,@#MMVEC   ;LOAD MEM MGMT ERROR VECTOR
004750  012737  000137  017076          MOV     #137,@#PUI2+76  ;000137,T43D =JMP  @#T43D
004756  012737  005006  017100          MOV     #T43D,@#PUI2+100
004764  005037  017102                  CLR     @#PUI2+102
004770  012767  170000  173000          MOV     #UM+PUM,PSW
004776  005237  177572                  INC     @#SR0           ;ENABLE MEMORY MGMT
005002  000137  040076                  JMP     @#UI2+76        ;GO DO INSTRUCTION
        005006                          RETURN=.
        017076                          .=PUI2+76
017076  000137  005006          T43A:   JMP     @#T43D
017102  000000                  T43B:   HLT                     ;ERROR: JMP FAILED
        005006                          .=RETURN
005006  000000                  T43D:   HLT                     ;ERROR: FAILED TO ABORT
005010                          T43C:
005010  022767  040145  172554          CMP     #PLA+UPG+VS2+1,SR0      ;CHECK SR0 (ABORT CONDITIONS
005016  001401                          BEQ     .+4             ;& FAILING PAGE #)
005020  000000                          HLT                     ;ERROR: INCORRECT ABORT CONDITIONS OR PAGE IDENT
005022  022767  040076  172546          CMP     #UI2+76,SR2     ;CHECK CONTENTS OF SR2
005030  001401                          BEQ     .+4             ;(PC OF ABORTED INSTRUCTION)
005032  000000                          HLT                     ;ERROR: INCORRECT PC IN SR2
005034  104000                          SCOPE                   ;SCOPE STORES PC IN R1 & SETS ALL STACK PTRS

                                ;CHECK ABORT AT DST12
                                ;WHEN INSTRUCTION FETCHES ADDRESS OF DEST', OPERAND', (UI5+4)
005036  012737  005106  000250          MOV     #T44C,@#MMVEC   ;LOAD MEM MGMT ERROR VECTOR
005044  012767  170000  172724          MOV     #UM+PUM,PSW     ;USER MODE!!!,PREV USER MODE!!
005052  012706  000600                  MOV     #UPTR,USP       ;SET USER STACK PTR
005056  012703  120006                  MOV     #UI5+6,R3
005062  012737  177776  017204          MOV     #-2,@#PUI5+4
005070  012737  004753  017200          MOV     #4753,@#PUI5    ;004753 = JSR 7,@-(R3)
005076  005237  177572                  INC     @#SR0           ;ENABLE MEMORY MGMT
005102  000137  120000                  JMP     @#UI5           ;GO DO INST,
        005106                          RETURN=.
```

```
                017200                          .=PUI5
017200  004753                  T44A:  JSR     7,@-(R3)
017202  000000                  T44B:  HLT                             ;ERROR!
                005106                  .=RETURN
005106  022706  001074          T44C:  CMP     #KPTR-4,KSP             ;CHECK STACK PTR
005112  001401                         BEQ     .+4
005114  000000                         HLT                            ;INCORRECT STACK PTR
005116  022767  140157  172446        CMP     #NRA+PLA+UPG+VS7+1,SR0  ;CHECK SR0 (ABORT CONDITIONS
005124  001401                         BEQ     .+4                    ;& FAILING PAGE #)
005126  000000                         HLT                            ;ERROR! INCORRECT ABORT CONDITIONS OR PAGE IDENT
005130  022767  120000  172440        CMP     #UI5,SR2                ;CHECK CONTENTS OF SR2
005136  001401                         BEQ     .+4                    ;(PC OF ABORTED INSTRUCTION)
005140  000000                         HLT                            ;ERROR! INCORRECT PC IN SR2
005142  106506                         MFPD    USP                    ;GET USER STACK PTR (ON KERNEL STACK)
005144  022716  000576                 CMP     #UPTR-2,(KSP)          ;CHECK THAT USER STACK DID NOT
005150  001401                         BEQ     .+4                    ;GET PUSHED
005152  000000                         HLT                            ;ERROR!
005154  022703  120004                 CMP     #UI5+4,R3              ;CHECK AUTO-DEC
005160  001401                         BEQ     .+4
005162  000000                         HLT                            ;ERROR!
005164  104000                         SCOPE                          ;SCOPE STORES PC IN R1 & SETS ALL STACK PTRS

                                ;CHECK ABORT AT DST10
                                ;WHEN INSTRUCTION FETCHES DESTINATION OPERAND (UIPDR5)
005166  012737  005240  000250         MOV     #T45C,@#MMVEC          ;LOAD MEM MGMT ERROR VECTOR
005174  012767  170000  172574         MOV     #UM+PUM,PSW            ;USER MODE!!!,PREV USER MODE!!
005202  012706  000600                 MOV     #UPTR,USP              ;SET USER STACK PTR
005206  005016                         CLR     (USP)
005210  012737  012667  017200         MOV     #012667,@#PUI5         ;012667,057606 = MOV  (USP)+,UIPDR5
005216  012737  057606  017202         MOV     #57606,@#PUI5+2        ;INSTRUCTION
005224  005037  017204                 CLR     @#PUI5+4
005230  005237  177572                 INC     @#SR0                  ;ENABLE MEMORY MGMT
005234  000137  120000                 JMP     @#UI5
                005240                  RETURN=.
                017200                  .=PUI5
017200  012667  057606          T45A:  MOV     (USP)+,UIPDR5=UI5+PUI5
017204  000000                  T45B:  HLT                             ;ERROR! FAILED TO ABORT
                005240                  .=RETURN

005240                          T45C:
005240  022767  140157  172324         CMP     #NRA+UPG+PLA+VS7+1,SR0  ;CHECK SR0 (ABORT CONDITIONS
005246  001401                         BEQ     .+4                    ;& FAILING PAGE #)
005250  000000                         HLT                            ;ERROR! INCORRECT ABORT CONDITIONS OR PAGE IDENT
005252  022767  120000  172316         CMP     #UI5,SR2                ;CHECK CONTENTS OF SR2
005260  001401                         BEQ     .+4                    ;(PC OF ABORTED INSTRUCTION)
005262  000000                         HLT                            ;ERROR! INCORRECT PC IN SR2
005264  005037  177572                 CLR     @#SR0                  ;DISABLE MEMORY MGMT
005270  005737  177612                 TST     @#UIPDR5
005274  001001                         BNE     .+4
005276  000000                         HLT
005300  104000                         SCOPE                          ;SCOPE STORES PC IN R1 & SETS ALL STACK PTRS
```

```
                                        ;CHECK ABORT AT JSR01
                                        ;ABORTS WHEN REGISTER (R5) IS PUSHED ON USER STACK
005302  012737  005352  000250            MOV     #T50C,@#MMVEC   ;LOAD MEM MGMT ERROR VECTOR
005310  012767  170000  172460            MOV     #UM+PUM,PSW     ;USER MODE!!!,PREV USER MODE!!
005316  012706  100000                    MOV     #U14,USP        ;SET USER STACK PTR
005322  005037  017276                    CLR     @#PU14-2
005326  005005                            CLR     R5
005330  012767  005350  173646            MOV     #T50D,TEMP
005336  005237  177572                    INC     @#SR0           ;ENABLE MEMORY MGMT
005342  004577  173636          T50A:     JSR     5,@TEMP         ;NON-RES ABORT AT JSR01
005346  000000          T50B:     HLT                             ;JSR FAILED & DID NOT ABORT
005350  000000          T50D:     HLT                             ;ERROR! FAILED TO ABORT
005352                  T50C:
005352  022767  140147  172212            CMP     #NRA+PLA+UPG+VS3+1,SR0  ;CHECK SR0 (ABORT CONDITIONS
005360  001401                            BEQ     .+4             ;& FAILING PAGE #)
005362  000000                            HLT                     ;ERROR! INCORRECT ABORT CONDITIONS OR PAGE IDENT
005364  022767  005342  172204            CMP     #T50A,SR2       ;CHECK CONTENTS OF SR2
005372  001401                            BEQ     .+4             ;(PC OF ABORTED INSTRUCTION)
005374  000000                            HLT                     ;ERROR! INCORRECT PC IN SR2
005376  106506                            MFPD    USP             ;PUSH USER STACK PTR ONTO KERNEL STACK
005400  022716  077776                    CMP     #U14-2,(KSP)    ;CHECK THAT USER STACK PTR DEC-
005404  001401                            BEQ     .+4             ;REMENTED
005406  000000                            HLT                     ;ERROR!
005410  005705                            TST     R5
005412  001401                            BEQ     .+4
005414  000000                            HLT
005416  104000                            SCOPE                   ;SCOPE STORES PC IN R1 & SETS ALL STACK PTRS

                                        ;CHECK ABORT AT TRP11
                                        ;ABORTS WHEN STATUS IS PUSHED ONTO USER STACK
005420  012737  005464  000250            MOV     #T52C,@#MMVEC   ;LOAD MEM MGMT ERROR VECTOR
005426  012767  140000  172366            MOV     #UM,IOTVEC+2
005434  012767  005462  172356            MOV     #T52D,IOTVEC
005442  012767  170000  172326            MOV     #UM+PUM,PSW     ;USER MODE!!!,PREV USER MODE!!
005450  005006                            CLR     USP             ;SET USER STACK PTR
005452  005237  177572                    INC     @#SR0           ;ENABLE MEMORY MGMT
005456  000004          T52A:     IOT                             ;NON-RESIDENT ABORT AT TRP11
005460  000000          T52B:     HLT                             ;ERROR! IOT & ABORT FAILED
005462  000000          T52D:     HLT                             ;ERROR! ABORT FAILED
005464  022706  001074  T52C:     CMP     #KPTR+4,KSP     ;CHECK STACK PTR
005470  001401                            BEQ     .+4
005472  000000                            HLT                     ;INCORRECT STACK PTR
005474  022766  170000  000002            CMP     #UM+PUM,2(KSP)  ;CHECK THAT CORRECT STATUS
005502  001401                            BEQ     .+4             ;WAS SAVED ON THE STACK
005504  000000                            HLT                     ;ERROR! INCORRECT STATUS
005506  022767  140157  172056            CMP     #NRA+UPG+PLA+VS7+1,SR0  ;CHECK SR0 (ABORT CONDITIONS
005514  001401                            BEQ     .+4             ;& FAILING PAGE #)
005516  000000                            HLT                     ;ERROR! INCORRECT ABORT CONDITIONS OR PAGE IDENT
005520  022767  005456  172050            CMP     #T52A,SR2       ;CHECK CONTENTS OF SR2
005526  001401                            BEQ     .+4             ;(PC OF ABORTED INSTRUCTION)
005530  000000                            HLT                     ;ERROR! INCORRECT PC IN SR2
005532  122737  000060  177777            CMPB    #60,@#PSW+1     ;CHECK FOR CORRECT PSW ON ABORT
005540  001401                            BEQ     .+4             ;(KM+PUM IN HIGH BYTE)
005542  000000                            HLT                     ;ERROR! INCORRECT PSW AFTER ABORT
```

```
005544  012737  030000  177776          MOV     #KM+PUM,@#PSW   ;KERNEL MODE!!! PREV SUPER MODE!!
005552  106506                          MFPD    USP             ;PUSH USER STACK PTR ONTO KERNEL STACK
005554  022716  177776                  CMP     #0-2,(KSP)      ;CHECK PUSHES
005560  001401                          BEQ     .+4
005562  000000                          HLT                     ;ERROR!
005564  104000                          SCOPE                   ;SCOPE STORES PC IN R1 & SETS ALL STACK PTRS

                                ;CHECK ABORT AT TRP14
                                ;ABORTS WHEN RETURN PC IS PUSHED ONTO USER STACK
005566  012737  005634  000250          MOV     #T53C,@#MMVEC   ;LOAD MEM MGMT ERROR VECTOR
005574  012767  170000  172174          MOV     #UM+PUM,PSW     ;USER MODE, PREV, USER MODE
005602  012706  040002                  MOV     #UI2+2,USP              ;SET USER STACK PTR
005606  012767  005632  172204          MOV     #T53D,IOTVEC
005614  012767  140340  172200          MOV     #UM+PRTY7,IOTVEC+2
005622  005237  177572                  INC     @#SR0           ;ENABLE MEMORY MGMT
005626  000004                  T53A:   IOT                     ;NON-RESIDENT ABORT AT TRP14
005630  000000                  T53B:   HLT                     ;ERROR! IOT & ABORT FAILED
005632  000000                  T53D:   HLT                     ;ERROR! ABORT FAILED
005634  022706  001074          T53C:   CMP     #KPTR-4,KSP     ;CHECK STACK PTR
005640  001401                          BEQ     .+4
005642  000000                          HLT                     ;INCORRECT STACK PTR
005644  022716  005630                  CMP     #T53B,(KSP)     ;CHECK RETURN PC
005650  001401                          BEQ     .+4
005652  000000                          HLT
005654  022766  170000  000002          CMP     #UM+PUM,2(KSP)  ;CHECK THAT CORRECT STATUS
005662  001401                          BEQ     .+4             ;WAS SAVED ON THE STACK
005664  000000                          HLT                     ;ERROR! INCORRECT STATUS
005666  022767  040143  171676          CMP     #PLA+UPG+VS1+1,SR0      ;CHECK SR0 (ABORT CONDITIONS
005674  001401                          BEQ     .+4             ;& FAILING PAGE #)
005676  000000                          HLT                     ;ERROR! INCORRECT ABORT CONDITIONS OR PAGE IDENT
005700  022767  005626  171670          CMP     #T53A,SR2       ;CHECK CONTENTS OF SR2
005706  001401                          BEQ     .+4             ;(PC OF ABORTED INSTRUCTION)
005710  000000                          HLT                     ;ERROR! INCORRECT PC IN SR2
005712  012767  030000  172056          MOV     #KM+PUM,PSW
005720  106506                          MFPD    USP             ;PUSH USER STACK PTR ONTO KERNEL STACK
005722  022716  037776                  CMP     #UI2-2,(KSP)    ;CHECK THAT USER STACK PTR WAS
005726  001401                          BEQ     .+4             ;DECREMENTED BY 4
005730  000000                          HLT                     ;ERROR!
005732  005067  172064                  CLR     IOTVEC+2
005736  012767  000022  172054          MOV     #IOTVEC+2,IOTVEC
005744  104000                          SCOPE                   ;SCOPE STORES PC IN R1 & SETS ALL STACK PTRS

                                ;CHECK ABORT AT FET03
005746  012737  006010  000250          MOV     #T56C,@#MMVEC   ;LOAD MEM MGMT ERROR VECTOR
005754  012767  170000  172014          MOV     #UM+PUM,PSW
005762  012706  000600                  MOV     #UPTR,USP
005766  012746  140000                  MOV     #UM,-(USP)
005772  012746  040100                  MOV     #UI2+100,-(USP)
005776  005037  017100                  CLR     @#PUI2+100
006002  005237  177572                  INC     @#SR0           ;ENABLE MEMORY MGMT
006006  000002                  T56B:   RTI
        006010                          RETURN=.
        017100                          .=PUI2+100
017100  000000                  T56A:   HLT                     ;ERROR! FAILED TO ABORT AT FET03
```

```
                006010                          ,=RETURN
006010  022706  001074          T56C:  CMP     #KPTR=4,KSP      ;CHECK STACK PTR
006014  001401                         BEQ     .+4
006016  000000                         HLT
006020  022767  040145  171544          CMP     #PLA+UPG+VS2+1,SR0      ;CHECK SR0 (ABORT CONDITIONS
006026  001401                         BEQ     .+4              ;& FAILING PAGE #)
006030  000000                         HLT                      ;ERROR: INCORRECT ABORT CONDITIONS OR PAGE IDENT
006032  022767  006006  171536          CMP     #T56B,SR2        ;CHECK CONTENTS OF SR2
006040  001401                         BEQ     .+4              ;(PC OF ABORTED INSTRUCTION)
006042  000000                         HLT                      ;ERROR: INCORRECT PC IN SR2
006044  104000                         SCOPE                    ;SCOPE STORES PC IN R1 & SETS ALL STACK PTRS

                                ;CHECK ABORT AT FET02
006046  012767  170000  171722          MOV     #UM+PUM,PSW
006054  012737  006106  000250          MOV     #T60C,@#MMVEC    ;LOAD MEM MGMT ERROR VECTOR
006062  012737  000005  017076          MOV     #5,@#PUI2+76     ;5 IS A RESET INSTRUCTION
006070  005037  017100                  CLR     @#PUI2+100
006074  005005                         CLR     R5
006076  005237  177572                  INC     @#SR0            ;ENABLE MEMORY MGMT
006102  000137  040076                  JMP     @#UI2+76         ;GO EXECUTE RESET
                006106                  RETURN=.
                017076                  .=PUI2+76
017076  000005                 T60A:  RESET                    ;ABORTS WHEN NEXT INST' FETCHED
017100  000000                         HLT                      ;ERROR: FAILED TO ABORT
                006106                  .=RETURN

006106                          T60C:
006106  022767  040145  171456          CMP     #PLA+UPG+VS2+1,SR0      ;CHECK SR0 (ABORT CONDITIONS
006114  001401                         BEQ     .+4              ;& FAILING PAGE #)
006116  000000                         HLT                      ;ERROR: INCORRECT ABORT CONDITIONS OR PAGE IDENT
006120  022767  040076  171450          CMP     #UI2+76,SR2      ;CHECK CONTENTS OF SR2
006126  001401                         BEQ     .+4              ;(PC OF ABORTED INSTRUCTION)
006130  000000                         HLT                      ;ERROR: INCORRECT PC IN SR2
006132  104000                         SCOPE                    ;SCOPE STORES PC IN R1 & SETS ALL STACK PTRS


                                ;CHECK ABORT AT
                                ;ABORTS WHEN INST FOLLOWING MARK IS FETCHED
006134  012767  170000  171634          MOV     #UM+PUM,PSW
006142  012737  006172  000250          MOV     #T63C,@#MMVEC    ;LOAD MEM MGMT ERROR VECTOR
006150  012746                         MOV     (PC)+,-(USP)     ;PUSH MARK INST ON USER STACK
006152  006401                         MARK    1                ;PUSH THIS  INST ON USER STACK
006154  012705  040100                  MOV     #UI2+100,R5      ;AFTER MARK EXECUTE INST AT T63A
006160  005037  017100                  CLR     @#T63A           ;WHICH IS A HALT
006164  005237  177572                  INC     @#SR0            ;ENABLE MEMORY MGMT
006170  000116                         JMP     (USP)            ;GO EXECUTE MARK AT SPTR-2
                006172                  RETURN=.
                017100                  .=PUI2+100
017100  000000                 T63A:  HLT                      ;SEG ABORT WHEN THIS INST, FETCHED AT
                006172                  .=RETURN
006172  022706  001074          T63C:  CMP     #KPTR=4,KSP      ;CHECK STACK PTR
006176  001401                         BEQ     .+4
006200  000000                         HLT
006202  022767  040145  171362          CMP     #PLA+UPG+VS2+1,SR0      ;CHECK SR0 (ABORT CONDITIONS
```

```
006210  001401                          BEQ     .+4                 ;& FAILING PAGE #)
006212  000000                          HLT                         ;ERROR! INCORRECT ABORT CONDITIONS OR PAGE IDENT
006214  022767  000576  171354          CMP     #UPTR-2,SR2         ;CHECK CONTENTS OF SR2
006222  001401                          BEQ     .+4                 ;(PC OF ABORTED INSTRUCTION)
006224  000000                          HLT                         ;ERROR! INCORRECT PC IN SR2
006226  106506                          MFPD    USP                 ;PUSH USER STACK PTR ONTO KERNEL STACK
006230  022716  000604                  CMP     #UPTR+4,(KSP)       ;CHECK USER STACK PTR
006234  001401                          BEQ     .+4
006236  000000                          HLT                         ;ERROR! INCORRECT USER STACK PTR
006240  023705  000602                  CMP     @#UPTR+2,R5         ;CHECK CONTENTS OF R5
006244  001401                          BEQ     .+4
006246  000000                          HLT                         ;ERROR!
006250  104000                          SCOPE                       ;SCOPE STORES PC IN R1 & SETS ALL STACK PTRS

                                ;CHECK ABORT AT TST,10
                                ;ABORTS WHEN INST FOLLOWING TST IS FETCHED
006252  012737  006306  000250          MOV     #T64C,@#MMVEC       ;LOAD MEM MGMT ERROR VECTOR
006260  012702  177777                  MOV     #-1,R2              ;R2=STATUS WORD ADDRESS (ODD BYTE)
006264  012737  105722  016676          MOV     #105722,@#KI0-2     ;105722=TSTB (R2)+
006272  005037  016700                  CLR     @#KI0
006276  005237  177572                  INC     @#SR0               ;ENABLE MEMORY MGMT
006302  000137  016676                  JMP     @#KI0-2             ;GO EXECUTE INSTRUCTION
        006306                          RETURN=.
        016676                          .=KI0-2
016676  105722                  T64A:   TSTB    (R2)+               ;ABORTS WHEN NEXT INST' IS FETCHED
016700  000000                  T64B:   HLT                         ;ERROR! FAILED TO ABORT
        006306                          .=RETURN
006306                          T64C:
006306  022767  040001  171256          CMP     #PLA+VS0+1,SR0      ;CHECK SR0 (ABORT CONDITIONS
006314  001401                          BEQ     .+4                 ;& FAILING PAGE #)
006316  000000                          HLT                         ;ERROR! INCORRECT ABORT CONDITIONS OR PAGE IDENT
006320  022767  016676  171250          CMP     #T64A,SR2           ;CHECK CONTENTS OF SR2
006326  001401                          BEQ     .+4                 ;(PC OF ABORTED INSTRUCTION)
006330  000000                          HLT                         ;ERROR! INCORRECT PC IN SR2
006332  005702                          TST     R2                  ;CHECK AUTO-INC
006334  001401                          BEQ     .+4
006336  000000                          HLT                         ;ERROR! AUTO-INC FAILED
006340  104000                          SCOPE                       ;SCOPE STORES PC IN R1 & SETS ALL STACK PTRS

                                ;CHECK ABORT AT D00,90
                                ;ABORTS WHEN INSTRUCTION FOLLOWING MOVB IS FETCHED
006342  012737  006400  000250          MOV     #T66C,@#MMVEC       ;LOAD MEM MGMT ERROR VECTOR
006350  012703  016700                  MOV     #KI0,R3
006354  005013                          CLR     (3)                 ;SET UP CODE (HALT)
006356  012743                          MOV     (7)+,-(R3)
006360  114203                          MOVB    -(R2),R3            ;THIS INSTRUCTION IS NOT EXECUTED
006362  012702  001204                  MOV     #TEMP,R2
006366  012722  100000                  MOV     #100000,(R2)+
006372  005237  177572                  INC     @#SR0               ;ENABLE MEMORY MGMT
006376  000113                          JMP     (R3)                ;GO EXECUTE MOVB INSTRUCTION
        006400                          RETURN=.
        016676                          .=KI0-2
016676  114203                  T66A:   MOVB    -(R2),R3            ;ABORTS WHEN THE NEXT INST IS FETCHED
016700  000000                  T66B:   HLT                         ;ERROR! FAILED TO ABORT HERE
```

```
                006400                          ;*RETURN
006400                                  T66C:
006400  022767  040001  171164                   CMP     #PLA+VS0+1,SR0  ;CHECK SR0 (ABORT CONDITIONS
006406  001401                                  BEQ     .+4             ;& FAILING PAGE #)
006410  000000                                  HLT                     ;ERROR: INCORRECT ABORT CONDITIONS OR PAGE IDENT
006412  022767  016676  171156                   CMP     #T66A,SR2       ;CHECK CONTENTS OF SR2
006420  001401                                  BEQ     .+4             ;(PC OF ABORTED INSTRUCTION)
006422  000000                                  HLT                     ;ERROR: INCORRECT PC IN SR2
006424  022703  177600                          CMP     #177600,R3      ;MOVB TO A REGISTER EXTENDS
006430  001401                                  BEQ     .+4             ;THE SIGN
006432  000000                                  HLT                     ;ERROR: INCORRECT RESULT IN R3
006434  104000                                  SCOPE                   ;SCOPE STORES PC IN R1 & SETS ALL STACK PTRS

                                        ;CHECK ACCESS VIOLATION ABORT
                                        ;ABORTS WHEN SOURCE DATA IS FETCHED USING DATIP WITH DEST ADDRESS READ ONLY
006436  012737  006470  000250                   MOV     #T72C,@#MMVEC   ;LOAD MEM MGMT ERROR VECTOR
006444  112737  000002  172314                   MOVB    #RDO,@#KIPDR6   ;SET KERNEL ADDRESS 140000-140077
                                                                        ;READ ABORT ON WRITE
006452  005037  016700                          CLR     @#PKI6          ;CLEAR CORRESPONDING PHYSICAL ADDRESS
006456  005237  177572                          INC     @#SR0           ;ENABLE MEMORY MGMT
006462  000261                                  SEC                     ;SET 'C'
006464  005537  140000          T72A:           ADC     @#KI6           ;ABORTS WHEN DATA IS FETCHED USING DATIP

006470                                  T72C:
006470  022767  020015  171074                   CMP     #AVA+VS6+1,SR0  ;CHECK SR0 (ABORT CONDITIONS
006476  001401                                  BEQ     .+4             ;& FAILING PAGE #)
006500  000000                                  HLT                     ;ERROR: INCORRECT ABORT CONDITIONS OR PAGE IDENT
006502  022767  006464  171066                   CMP     #T72A,SR2       ;CHECK CONTENTS OF SR2
006510  001401                                  BEQ     .+4             ;(PC OF ABORTED INSTRUCTION)
006512  000000                                  HLT                     ;ERROR: INCORRECT PC IN SR2
006514  022766  000001  000002                   CMP     #C,2(KSP)       ;CHECK THAT CORRECT STATUS
006522  001401                                  BEQ     .+4             ;WAS SAVED ON THE STACK
006524  000000                                  HLT                     ;ERROR: INCORRECT STATUS
006526  005037  177572                          CLR     @#SR0           ;DISABLE MEMORY MGMT
006532  005737  016700                          TST     @#PKI6          ;CHECK THAT ADDRESS WAS NOT WRITTEN
006536  001401                                  BEQ     .+4
006540  000000                                  HLT                     ;ERROR: DATA WRITTEN INTO READ ONLY ADDRESS
006542  104000                                  SCOPE

                                        ;CHECK ACCESS VIOLATION ABORT
                                        ;ABORTS WHEN SOURCE DATA IS FETCHED FROM READ ONLY SPACE USING A DATIP,
006544  012737  006570  000250                   MOV     #T73C,@#MMVEC   ;LOAD MEM MGMT ERROR VECTOR
006552  005037  016700                          CLR     @#PKI6          ;PRESET ADDRESS
006556  005237  177572                          INC     @#SR0           ;ENABLE MEMORY MGMT
006562  000261                                  SEC                     ;SET 'C'
006564  106037  140001          T73A:           RORB    @#KI6+1         ;ABORTS WHEN RESULT IS WRITTEN
006570                                  T73C:
006570  022767  020015  170774                   CMP     #AVA+VS6+1,SR0  ;CHECK SR0 (ABORT CONDITIONS
006576  001401                                  BEQ     .+4             ;& FAILING PAGE #)
006600  000000                                  HLT                     ;ERROR: INCORRECT ABORT CONDITIONS OR PAGE IDENT
006602  022767  006564  170766                   CMP     #T73A,SR2       ;CHECK CONTENTS OF SR2
006610  001401                                  BEQ     .+4             ;(PC OF ABORTED INSTRUCTION)
006612  000000                                  HLT                     ;ERROR: INCORRECT PC IN SR2
006614  022766  000001  000002                   CMP     #C,2(KSP)       ;CHECK THAT CORRECT STATUS
```

```
006622  001401                          BEQ     .+4             ;WAS SAVED ON THE STACK
006624  000000                          HLT                     ;ERROR! INCORRECT STATUS
006626  005037  177572                  CLR     @#SR0           ;DISABLE MEMORY MGMT
006632  005737  016700                  TST     @#PKI6
006636  001401                          BEQ     .+4
006640  000000                          HLT                     ;ERROR! ADDRESS WAS WRITTEN
006642  012737  000006  172314          MOV     #6,@#KIPDR6     ;SET KDPDR R/W
006650  104000                          SCOPE                   ;SCOPE STORES PC IN R1 & SETS ALL STACK PTRS


                                ;CHECK ABORT AT SVC,80 USING 'T' BIT TRAP
                                ;CHECK ABORT WHEN PSW IS NON-RESIDENT
006652  012737  006706  000250          MOV     #T102C,@#MMVEC  ;LOAD MEM MGMT ERROR VECTOR
006660  005002                          CLR     R2              ;PRESET DESTINATION
006662  005237  177572                  INC     @#SR0           ;ENABLE MEMORY MGMT
006666  012746  140017                  MOV     #UM+17,-(KSP)   ;'NEW' STATUS ON STACK
006672  012746  006700                  MOV     #.+6,-(KSP)     ;RETURN PC
006676  000002                          RTI                     ;SET STATUS AND EXECUTE NEXT INST,
006700  013702  177776          T102A:  MOV     @#PSW,R2        ;PSW IS NON-RESIDENT IN USER MODE
006704  000000                          HLT                     ;ERROR! FAILED TO ABORT
006706                          T102C:
006706  022767  140157  170656          CMP     #NRA+PLA+UPG+VS7+1,SR0  ;CHECK SR0 (ABORT CONDITIONS
006714  001401                          BEQ     .+4             ;& FAILING PAGE #)
006716  000000                          HLT                     ;ERROR! INCORRECT ABORT CONDITIONS OR PAGE IDENT
006720  022767  006700  170650          CMP     #T102A,SR2      ;CHECK CONTENTS OF SR2
006726  001401                          BEQ     .+4             ;(PC OF ABORTED INSTRUCTION)
006730  000000                          HLT                     ;ERROR! INCORRECT PC IN SR2
006732  022766  140017  000002          CMP     #UM+17,2(KSP)   ;CHECK THAT CORRECT STATUS
006740  001401                          BEQ     .+4             ;WAS SAVED ON THE STACK
006742  000000                          HLT                     ;ERROR! INCORRECT STATUS
006744  005702                          TST     R2              ;CHECK THAT R2 WAS NOT LOADED
006746  001401                          BEQ     .+4
006750  000000                          HLT                     ;ERROR! DEST (R2) WAS CHEANGED
006752  104000                          SCOPE

006754  005267  172220          END:    INC     ICNT
006760  022767  005000  172212          CMP     #5000,ICNT
006766  001402                          BEQ     DONE
006770  000167  172224                  JMP     BEGIN
006774  012767  000007  170564  DONE:   MOV     #7,TPB
007002  105767  170556                  TSTB    TPS
007006  100375                          BPL     .-4
007010  013702  000042                  MOV     @#42,X2         ;GET DECTAPE MONITOR RETURN ADDRESS
007014  001404                          BEQ     DONE1           ;DO NOT RETURN TO MON IF (42)=0
007016  004712                          JSR     7,(2)           ;RETURN TO DECTAPE MONITOR
007020  000240                          NOP                     ;ACT11
007022  000240                          NOP                     ;OVERLAY
007024  000240                          NOP                     ;AREA
007026  000167  172160          DONE1:  JMP     START

        000001                          .END
```

```
AVA     = 020000        BEGIN   001220        BIT13   = 020000        BIT14   = 040000
BIT15   = 100000        BIT6    = 000100        BIT8    = 000400        C       = 000001
DM      = 000400        DONE    006774        DONE1   007026        DWN     = 000010
ED      = 000010        EMTVEC  = 000030        END     006754        ENMM    = 000001
ERRVEC  = 000010        FPVEC   = 000244        HLT     = 000000        ICNT    001200
IOTVEC  = 000020        IS      = 000000        KIPAR0  = 172340        KIPAR1  = 172342
KIPAR2  = 172344        KIPAR3  = 172346        KIPAR4  = 172350        KIPAR5  = 172352
KIPAR6  = 172354        KIPAR7  = 172356        KIPDR0  = 172300        KIPDR1  = 172302
KIPDR2  = 172304        KIPDR3  = 172306        KIPDR4  = 172310        KIPDR5  = 172312
KIPDR6  = 172314        KIPDR7  = 172316        KI0     = 016700        KI6     = 140000
KM      = 000000        KPG     = 000000        KPTR    = 001100        KSP     =%000006
MMK     001320          MMVEC   = 000250        MM0     001232          N       = 000010
NRA     = 100000        NR0     = 000000        PC      =%000007        PFVEC   = 000024
PKI0    = 016600        PKI6    = 016700        PKM     = 000000        PLA     = 040000
PRTY4   = 000200        PRTY7   = 000340        PSW     = 177776        PUI1    = 017100
PUI2    = 017000        PUI3    = 017400        PUI4    = 017300        PUI5    = 017200
PUM     = 030000        RDO     = 000002        RETURN  = 006400        RW      = 000006
RWT     = 000004        R0      =%000000        R1      =%000001        R2      =%000002
R3      =%000003        R4      =%000004        R5      =%000005        SCOPE   = 104000
SCOPEA  000442          SCOPEX  000476        SHLT    000400          SHLTA   000436
SP      =%000006        SR0     = 177572        SR0T    001202          SRI     = 177574
SR2     = 177576        START   001212        SWR     = 177570        T       = 000020
TBITVE  = 000014        TEMP    = 001204        TKB     = 177562        TKS     = 177560
TPB     = 177566        TPS     = 177564        TPVEC   = 000064        TRAPVE  = 000034
T0      001454          T0A     001504        T0B     001506          T0C     001510
T1A     001610          T1B     001612        T1C     001614          T10A    002434
T10B    002440          T10C    002442        T102A   006700          T102C   006706
T13A    002536          T13B    002540        T13C    002544          T13D    002542
T14A    002664          T14B    002666        T14C    002672          T14D    002670
T16A    002762          T16B    002764        T16C    002770          T16D    002766
T2A     001704          T2B     001706        T2C     001710          T20A    016676
T20B    016702          T20C    003112        T21A    016676          T21B    016702
T21C    003232          T22     016674        T22A    016676          T22AA   016700
T22B    016702          T22C    003332        T24A    003434          T24B    003436
T24C    003440          T25A    003532        T25B    003534          T25C    003536
T3A     002010          T3B     002012        T3C     002014          T30A    003620
T30C    003624          T31A    003742        T31B    003744          T31C    003746
T32A    004054          T32B    004056        T32C    004060          T33A    004166
T33B    004170          T33C    004172        T35A    004260          T35B    004264
T35C    004266          T36A    004364        T36B    004366          T36C    004370
T4A     002160          T4B     002162        T4C     002164          T40A    017074
T40B    017102          T40C    004504        T41A    017200          T41B    017202
T41C    004600          T42A    004670        T42B    004672          T42C    004674
T43A    017076          T43B    017102        T43C    005010          T43D    005006
T44A    017200          T44B    017202        T44C    005106          T45A    017200
T45B    017204          T45C    005240        T50A    005342          T50B    005346
T50C    005352          T50D    005350        T52A    005456          T52B    005460
T52C    005464          T52D    005462        T53A    005626          T53B    005630
T53C    005634          T53D    005632        T56A    017100          T56B    006006
T56C    006010          T6A     002300        T6B     002304          T6C     002306
T60A    017076          T60C    006106        T63A    017100          T63C    006172
T64A    016676          T64B    016700        T64C    006306          T66A    016676
T66B    016700          T66C    006400        T7A     002356          T7B     002362
T7C     002364          T72A    006464        T72C    006470          T73A    006564
```

```
T73C     006570          UIPAR0 = 177640          UIPAR1 = 177642          UIPAR2 = 177644
UIPAR3 = 177646          UIPAR4 = 177650          UIPAR5 = 177652          UIPAR6 = 177654
UIPAR7 = 177656          UIPDR0 = 177600          UIPDR1 = 177602          UIPDR2 = 177604
UIPDR3 = 177606          UIPDR4 = 177610          UIPDR5 = 177612          UIPDR6 = 177614
UIPDR7 = 177616          UI1    = 020000          UI2    = 040000          UI3    = 060000
UI4    = 100000          UI5    = 120000          UM     = 140000          UP     = 000000
UPC    = 000140          UPTR   = 000600          USP    =%000006          V      = 000002
VS0    = 000000          VS1    = 000002          VS2    = 000004          VS3    = 000006
VS4    = 000010          VS5    = 000012          VS6    = 000014          VS7    = 000016
W      = 000100          Z      = 000004          .      = 007032
```

ERRORS DETECTED:  0